GeoVu Tools

for Data Publishers

Reference Guide

UNITED STATES DEPARTMENT OF COMMERCE
National Oceanic and Atmospheric Administration
National Environmental Satellite, Data, and Information Service
National Geophysical Data Center
Boulder, Colorado 80303 USA

NOTICE

The information in this publication is subject to change without notice. NOAA reserves the right to revise this publication and to make changes in the content hereof without obligation of NOAA to notify any person of such revisions or changes.

DISCLAIMER

While every precaution has been taken in the preparation of this publication, NOAA assumes no responsibility for any errors or omissions that may appear herein. Nor does it assume any liability for damages resulting from the use of the information contained in this publication.

TRADEMARKS

All brand and product names are trademarks of their respective companies. Mention of a commercial company or product does not imply endorsement by NOAA or the Department of Commerce. Using information from this publication concerning proprietary products for publicity or advertising purposes is not authorized.

Contents

PREPARING DATA FOR USE 4
GeoVu Tools 4

FORMAT FILES 6
Writing Format Descriptions 6
FreeForm Variable Types 7
Headers 8
Format File Examples 8

EQUIVALENCE TABLES 11
Equivalence Table Syntax 11
name_equiv Section 12
constant Section 13
Example Equivalence Files 13

MENU FILES 15
Menu File Syntax 15
VOLUME_ID Section 16
INTRODUCTION Section 16
Special Sections 17
Menu Sections 17
Help Sections 22
Equivalence Sections 23
Format Sections 25
Example Format Sections 26
Header Section 30

DEFAULT SEARCH SEQUENCE 31
… 32

GEOVU KEYWORDS 33

MENU FILE UTILITIES 59
Checking Menu Files 59

APPENDIX A EXAMPLE MENU FILE 63
GRAVITY CD 63

1

Preparing Data For Use

GeoVu … developed at the National Geophysical Data Center (NGDC) …
More overview information to be provided …

GeoVu Tools

The first step in preparing data for use with GeoVu is to specify the format of the input data and of any associated data headers in a format description file. You must also create an equivalence table, which specifies the equivalence between display parameter names native to a data set with GeoVu keywords used to define display parameters.

For a data collection, whether on one or more CD-ROMs or diskettes, you should write a menu navigation file. You can optionally include the format and equivalence specifications as sections in a menu file instead of writing separate files. You can also optionally include equivalence specifications as sections in format description files.

Levels of compatibility

- format descriptions
- keyword equivalences
- menu files

Approach to writing accessory files that make data accessible …

2

Format Files

In order to work with a data file, GeoVu needs to have information about the data including format (e.g., binary) and variable type (e.g., long). You generally do not need to specially format or reprocess data for display in GeoVu, but the data formatting must be described in format files that are used by GeoVu as it runs. In some cases, GeoVu may be able to run in the absence of a format file by using defaults and information that the user types into dialogs during the session. You can include format descriptions in menu files rather than placing them in separate files (see chapter 4).

Writing Format Descriptions

Format descriptions define the formats of input and output data and headers. FreeForm, a suite of programs for data access and management developed at NGDC, provides a mechanism for describing data. GeoVu is a FreeForm-based application that uses format descriptions to correctly access data. For complete information about the FreeForm Data Access System, see the *FreeForm User's Guide*.
To write a format description, you must know which of the following types of data and headers you are working with:

1. binary, ASCII, or dBASE data
2. a file header in the data file
3. record headers in the data file
4. a file header stored in a file separate from the data
5. record headers stored in a file separate from the data
6. if there is a header, is it of fixed-length or variable position type

A description of output data must be provided only if the data will be converted by GeoVu using the Write To Disk view type.

FreeForm Variable Types

The data sets you produce and use may contain a variety of variable types. The characteristics of the variable types that FreeForm supports are summarized in the table below. (For a description of each variable type, see chapter 3 in the *FreeForm User's Guide*.) The sizes in the table are machine-dependent. Those given are for PC-compatible machines.

Table 1: Variable Types

Name	Minimum Value	Maximum Value	Binary Size (bytes)	Precision (significant digits)
char			*	
uchar	0	255	1	
short	-32,767	32,767	2	
ushort	0	65,535	2	
long	-2,147,483,647	2,147,483,647	4	
ulong	0	4,294,967,295	4	
float	10^{-37}	10^{38}	4	6
double	10^{-307}	10^{308}	8	15
constant			*	
initial			record length	
convert			*	

* User-specified

Headers

Headers provide important information about the structure of data files, and so they are usually searched and read by GeoVu. If you want GeoVu to access header information, you must prepare a specification of the header format as well as the data format.

There are several ways to define a header format and its association with the appropriate data format.

- ➔ Include a header format description along with the associated data format description in a format description file (**.fmt**) or format section in a menu file. For complete information about format description files, see the *FreeForm User's Guide*.
- ➔ Define the GeoVu keyword **ff_header_format** in an equivalence table.
- ➔ Use the header format file name as the header variable name inside the data format

Format File Examples

The format file for binary ERDAS data, **erdas_r.bfm**:

```
erdas_r.bfm 1 128 header 0
bil 1 1 uchar 0
```

The header format file for binary ERDAS data, **erd_bhdr.bfm**:

```
erdas_id 1 6 char 0
ipack 7 8 short 0
number_of_bands 9 10 short 0
number_of_columns  17 20 float 0
number_of_rows   21 24 float 0
rx              25 28 float 5
```

```
ry              29 32 float 5
map_projection  89 90 short   0
number_of_classes 91 92 short   0
left_map_x 113 116 float 5
upper_map_y 117 120 float 5
grid_size(x)      121 124 float 5
grid_size(y)      125 128 float 5
```

The format file for binary GeoVu data, **geovu_ra.bfm**:

Note! An **.eqv** file must define: **ff_header_format char geo_bhdr.bfm**

```
header 1 348 header 0
data 1 1 uchar 0
```

The format file for binary GeoVu headers, **geo_bhdr.bfm**:

```
 file_name 1 80 char 0
 file_title 81 160 char 0
 data_type 161 166 char 0
 data_representation 167 172 char 0
 number_of_rows 173 174 short 0
 number_of_columns 175 176 short 0
 number_of_bands 177 178 short 0
 rows_in_original 179 180 short 0
 columns_in_original 181 182 short 0
 number_of_arrays 183 184 short 0
 sequence_of_array 185 192 char 0
 data_value_unit 193 204 char 0
 data_scale_factor 205 208 float 0
 missing_flag 209 212 float 0
 maximum_data_value 213 216 float 0
 minimum_data_value 217 220 float 0
 grid_unit 221 228 char 0
 grid_size(y)  229 232 float 0
 grid_size(x) 233 236 float 0
 registration_of_grid 237 240 char 0
 upper_latitude 241 244 float 0
 lower_latitude 245 248 float 0
 left_longitude 249 252 float 0
 right_longitude 253 256 float 0
 map_projection 257 264 char 0
 central_meridian 265 268 float 0
 standard_latitude 269 272 float 0
 upper_map_y 273 276 float 0
 lower_map_y 277 280 float 0
 left_map_x 281 284 float 0
 right_map_x 285 288 float 0
 system_palette 289 292 char 0
 user_palette 293 296 char 0
 statistics_file 297 300 char 0
 annotation_file 301 304 char 0
 comments 305 344  char 0
 comment_file 345 348 char 0
```

The format file for binary IDRISI data, **idris_ra.bfm**:

```
IDRISI_HEADER 0 0 header 0
data 1 2 short 0
```

The format for the IDRISI-4 headers, file **idr_bhdr.afm**:

```
file%title  0 0 char 0
data%type   0 0 char 0
file%type   0 0 char 0
columns     0 0 short 0
rows        0 0 short 0
ref.%system 0 0 char 0
ref.%units  0 0 char 0
unit%dist.  0 0 double 7
min.%X      0 0 double 7
max.%X      0 0 double 7
min.%Y      0 0 double 7
max.%Y      0 0 double 7
pos'n%error 0 0 char 0
resolution  0 0 char 0
min.%value  0 0 short 0
max.%value  0 0 short 0
value%units 0 0 char 0
value%err   0 0 char 0
flag%value  0 0 char 0
flag%def'n  0 0 char 0
legend%cats 0 0 short 0
```

This description uses a position of 0 to alert the data access processing (FreeForm) that the headers have variable positioning. This is what an IDRISI4 header looks like. It is in a separate file (not in the data file):

```
file title  : KGALLO MONTHLY GVI :  DEC 1990
data type   : byte
file type   : binary
columns     : 2160
rows        : 1080
ref. system : lat/long
ref. units  : deg
unit dist.  : 0.1666667
min. X      : -180.0000000
max. X      : 180.0000000
min. Y      : -90.0000000
max. Y      : 90.0000000
pos'n error : unknown
resolution  : unknown
min. value  : 0
max. value  : 188
value units : unspecified
value error : unknown
flag value  : none
flag def'n  : none
legend cats : 0
```

3

Equivalence Tables

To display views of image data, a set of variables is required to describe aspects of the display such as the number of rows and the number of columns. These variables are often given in headers associated with the data files. GeoVu has its own set of names (keywords) for these variables. An equivalence table equates the variable names native to the data set with the GeoVu keywords. The equivalence table tells GeoVu what names are used in the headers of the data files. These equivalences are placed in **.eqv** files that can be kept with the data files.

The base file name of an **.eqv** file should be the same as the base file name of the data format file (not the header format file) and should be put in the same directory.When a menu (**.men**) file is written for a data collection, the equivalence table can be placed in that file as a **_eqv** section instead of in a separate **.eqv** file. See chapter 4 for details.

Equivalence Files

An equivalence file (**.eqv**) is an ASCII file that you create using a text editor. It is composed of sequences of two types of blocks: a **name_equiv** block and a **constant** block. The blocks begin with the literal string `begin name_equiv` or `begin constant` and end with `end name_equiv` or `end constant`. Each type of block can appear one or more times in any order. A **.eqv** file has a form similar to the following:

```
begin name_equiv
      .
      .
      .
  name_equiv block body
      .
      .
      .
end name_equiv
begin constant
      .
      .
      .
  constant block body
      .
      .
      .
end constant
```

name_equiv Section

The statements within this block define the association of GeoVu keywords and data-specific variable names, but also value relationships and constant values. The body of a **name_equiv** section includes two kinds of statements: name equivalence statements assigning header variable names to the GeoVu keywords and value equivalence statements defining the equivalence between values of GeoVu variables and user-defined variables.

The name equivalence statement begins with an identifier $, followed by a GeoVu variable name (no space between $ and the GeoVu name), space, and user-defined variable name. There is only one equivalence statement per line:

```
$GeoVu_Name User-defined-name
```

Example

GeoVu defines the size of an image by the variables **number_of_rows** and **number_of_columns**. But IDRISI defines it by variables named **rows** and **columns**. In order to access an IDRISI image file, **rows** and **columns** must be found in the IDRISI header file and the variable names **rows** and **columns** must be equated to the GeoVu keywords **number_of_rows** and **number_of_columns**. This equivalence is seen in the file **idrisi_ra.eqv**:

```
$number_of_rows rows
$number_of_columns columns
```

In some cases, a user-defined variable name may contain blanks (spaces). In order to avoid errors, blanks should be replaced by the character %. GeoVu replaces % with a space when it reads the name. For example, in IDRISI, the map projection for an image is called **refs. system** which is called **map_projection** in GeoVu. The equivalence statement is:

```
$map_projection  refs.%system
```

In some cases, the meaning of the values of a user-defined variable may be different from that of a GeoVu variable even when the names are the same. A value-equivalence statement should then be placed immediately after the name-equivalence statement which defines the variable name. This implies that even when a variable name is identical to a GeoVu name, the name-equivalence statement should be present if the meanings of their values are different. The value-equivalence statement has the form:

GeoVu_data_type GeoVu_value your_data_type your_value

Example of a name equivalence statement followed by a value equivalence statement:

```
$map_projection  refs.%system
      char lat/lon char lat./long.
```

Separators are single spaces. For values that contain blanks, use % to replace the blanks. The *..._data_type* descriptors that can be used are the type descriptors of FreeForm. A name equivalence statement can be followed by several value equivalence statements, or none, one statement per line.

constant Section

The body of the **constant** section consists of statements which define constants using the form:

constant_name constant_type constant_value

Separators are single spaces. If either the name or value contains blanks, use % to replace them. Both constant names and constant values should use user-defined variable names and user-defined values if the user-defined variable has a name equivalence statement in the equivalence table. Otherwise, use the GeoVu keywords and their corresponding values.

An example of a constant statement, defining **rows** to be 512:

```
      rows short 512
```

To retrieve the value of a GeoVu keyword, GeoVu first converts the keyword to the user-defined variable name according to the equivalence table statements. Then it looks for the value of the variable in the file header contents. If it can't be found, the program then searches the constant statements. So the value in a constant statement provides a default value for the variable.

Example Equivalence Files

The **.eqv** file used by ERDAS raster files, **erdas_ra.eqv**:

```
begin name_equiv
      $data_representation ipack
            char uchar short 0
            char short short 2
end name_equiv
begin constant
      data_type char image
      missing_flag short 0
      header_type char header_embedded
      image_format char bil
      ff_header_format char erd_bhdr.bfm
end constant
```

The **.eqv** file used by GeoVu raster files, **geovu_ra.eqv**:

```
begin constant
      data_type char image
      header_type char header_embedded
      image_format char bsq
      ff_header_format char geo_bhdr.bfm
end constant
```

The **.eqv** file used by IDRISI raster files, **idrisi_r.eqv**:

```
begin name_equiv
      $file_title image%title
      $data_representation data%type
            char uchar char byte
            char short  char integer
            char float char real
      $number_of_rows rows
      $number_of_columns columns
      $minimum_value minimum
      $maximum_value maximum
      $grid_size(x) cell%x
      $grid_size(y) cell%y
      $data_value_unit data%units
      $right_map_x max%X
      $left_map_x min%X
      $upper_map_y max%Y
      $lower_map_y min%Y
      $map_projection ref.%system
      $grid_unit ref.%units
      $comments comment
end name_equiv
begin constant
      ff_header_format char idr_bhdr.afm
      header_type char header_separated_varied
      data_type char image
      header_file_ext char .doc
      image_type char BSQ
      delimiter_item char \n
      _distance short 14
end constant
```

4

Menu Files

GeoVu uses custom menus that allow users to access the files on CDs or in data collections. The menu system presents and interprets descriptive text strings that provide file names and full directory locations to GeoVu. Thus users are not required to have an expert acquaintance with the format and organization of the data they are examining. Custom menus are derived from ASCII menu (**.men**) files written at NGDC or by other data providers. Each CD product or data collection has one corresponding menu file which is read by the GeoVu software during execution.

Menu File Syntax

A menu file is composed of sections similar to the following:

```
*VOLUME_ID
volume id name
*INTRODUCTION
descriptive text
*MAIN MENU
NULL
menu item name #item action
      .
      .
      .
*menu name
parent menu name
menu item name #item action
      .
      .
      .
*menu name_help
help text for menu_name
```

Each line beginning with * is a section title. Section titles must not start with spaces or tabs and cannot contain the hash character (#). The menu guide is case-sensitive, so names and keywords must be exact in the menu file. If the data collection for which you write a menu file will be used on both Unix and DOS systems, you should use lowercase letters in paths and file names.

To check a menu file for completeness, you can use the utility program **menutil**. Refer to chapter 6.

The names of CD products and data collections supported by GeoVu are listed in the accessory file **cdm_serv.lst** (**cdmenu.lst** prior to GeoVu release 4.0). This data source list file is managed within GeoVu by the Edit menu item Data Source List. It can be edited: add a line to this file naming each new CD and giving the corresponding menu file name.

VOLUME_ID Section

The VOLUME_ID section identifies the volume name of the CD which the menu file describes.

```
*VOLUME_ID
volume name
```

GeoVu reads the volume name from the CD in the CD-ROM reader, as selected in the Set Data Source dialog. GeoVu checks to see if the volume name on the CD matches that in the menu file. If it doesn't, the user is prompted to insert the correct volume into the CD-ROM drive.

Example

The VOLUME_ID section of the menu file for the CD containing K. Gallo's Global Vegetation Index:

```
*VOLUME_ID
KGALLO_GVI
```

If the CD product has multiple volumes, put the volume ID names together on one line separated by commas. Each menu file has only one VOLUME_ID section.

INTRODUCTION Section

The INTRODUCTION section (must be capitalized) contains text providing general information about the data collection. You can include as many lines of text as you want.

```
*INTRODUCTION
```

```
descriptive text
```

When the user selects a data source in GeoVu using the menu item Set Data Source**,** GeoVu displays the information in this section.

Example

The INTRODUCTION section of the menu file for the Global Vegetation Index:

```
*INTRODUCTION
                        GLOBAL CHANGE DATA BASE, VOLUME 2
                         Experimental Calibrated Global
                         Vegetation Index from NOAA AVHRR
                                    1985-1991

                    NOAA/NGDC Global Change Database Program

For more information please contact:

David A. Hastings,Remote Sensing and Data Integration
phone:(303) 497-6729
fax:  (303) 497-6513
```

Special Sections

MENU_NAME (formerly CD_MENU_NAME) is a section containing a string of up to 60 characters which is used as a logical title for all references to the current volume. The title appears in the Data Source dialog.

```
*MENU_NAME
Snow and Ice Data 1990
```

PALETTES_FILE is a section containing the name of a list file of custom color palettes. This file is read into GeoVu when the current menu file is in effect. The file should reside in the working directory defined by GEOVUDIR, but can include a full path. This section was added to the menu syntax after GeoVu release 4.0.

```
*PALETTES_FILE
ccapfiles.lst
```

Some CDs or data collections contain images appropriate for slide show animations. To activate GeoVu's slide show feature, include the ANIMATION and SLIDE SHOW sections with a value of TRUE.

```
*SLIDE SHOW
TRUE
*ANIMATION
TRUE
```

Menu Sections

Menu sections delineate hierarchical menu structure for purposes of menu navigation. They have the following form:

```
*menu name
parent menu name
menu item name  #item action
      .
      .
      .
```

The *parent menu name* is the name of an upper level menu. For a hierarchical menu system, the root menu must exist and it must be called MAIN MENU. The parent menu name for MAIN MENU must be NULL.

```
*MAIN MENU
NULL
menu item name  #item action
      .
      .
      .
```

Menu Items

For each menu section, you can have up to 400 menu items, though it is not recommended that you approach this maximum. Each menu item is placed on one text line and consists of the menu item name followed by an item action with # as a prefix. You can precede the # with a space, but do not insert a space between the # and the first letter of the item action to the right of it.
When GeoVu is running, the menu item names appear in selection lists within a dialog. The dialog usually presents a name of about 75 characters in width, although it is recommended that all menu item names be less than 60 characters.

Item Actions

The three types of item actions are submenu, file name, and terms file.

Submenu

Display the named submenu, which should also appear as a menu item name somewhere in the menu file.

*menu item name #*submenu_name*

Example

```
Global Vegetation Datasets #*Special Vegetation Datasets
```

File Name

Return a file name to GeoVu. (This is a leaf in the hierarchical menu tree.) The file name should include the path name without the drive name.

menu item name #*directory\filename*

Example

```
Special Vegetation Datasets #\special\data\gvdata.img
```

If the file is an ASCII file, prefix the name with $. This enables the DATA button–the viewing option available only for ASCII data (text) files. The menu system displays the contents of the file in a dialog if this button is selected.

menu item name #$*directory\filename*

Example

```
Special Vegetation Document #$\special\data\gvdata.doc
```

Terms File

Return the name of a terms file and display its contents. A terms file is an ASCII file that defines field names and code values. It has the following form:

```
*title1
descriptive text
*title2
descriptive text
     .
     .
     .
```

When the user selects a menu item that displays a terms file, the menu system presents a list of all titles found in the file. Once the user selects a title, the corresponding text is displayed. Use the following form in the menu file when the item action is a terms file:

menu item name ##filename

Example

```
Special Vegetation Text ##gvdata.txt
```

Example

```
Term definitions           ##MINTERMS.txt
```

The ## prefix is used with ASCII files found in the GeoVu working directory. The GeoVu working directory path will be appended to the file name.

The following is an example of a file, MINTERMS.TXT, written to provide a glossary of terms and then referenced in a GeoVu menu file. It is used in two ways.

Access to it through menu navigation is provided by creating a submenu called Term definitions:

```
*MAIN MENU
NULL
Survey Listings            #*Survey Listings
Gridded Data               #*Gridded Data (CD 02)
Term definitions           ##MINTERMS.TXT
```

The GeoVu keyword **term_definition** is defined in one of the **_eqv** sections of the menu file, using this file name so that FACTS buttons will find this text.

```
begin constant
    data_type char point
      map_projection char lat/lon
      term_definition char MINTERMS.TXT
    histogram_dir char l">_"
end constant
```

Example

An excerpt from **minterms.txt**:

```
*fiducial_number
A unique number corresponding to points of simultaneity; e.g. a mark on a magnetic
intensity record showing which point corresponds to a point on an altimeter record
and to a point on the map or navigation records which were made at the same time.

*flight_line_headers_and_fills
A fixed length header block to accommodate not more than 32 flight line header
records per data file.  Each flight-line header contains the following parameters:
     I4   flight-line number
     I6   logical record number that begins data for this flight-line
     I3   day of year
     F8.4 latitude of the ground monitor
     F8.4 longitude of the ground monitor
```

```
     51X  blanks to fill 80 byte record
     cr   carriage return byte (hex: 0D)
     lf   line feed byte (hex: 0A)
     ---
      82 bytes          ===> x 32 = 2624
```

NOTE: the fixed length header block is actually 2622 bytes since the last cr/lf bytes are not present.

*flight_number
Number assigned to a flight path corresponding to a specific data set.

*flight_tape_number
Number of a survey tape that corresponds to a particular flight segment.

*funding_agency
Agency that provides funds to a particular contractor.
*geologic_map_unit_surface_code
A combination of two four byte codes (2 A4) indicating the local surface geologic structure.

*gmt_time_of_day_hhmmss
Greenwich mean time - time of day referenced to the prime meridian in hours:minutes:seconds. Hours vary from 0-23; minutes and seconds vary from 0-59.
*julian_date
Day of year (e.g. 010 is January 10).
*latitude
This is the geographic latitude expressed as a decimal number. The units are degrees. The range is 0 to +90.0 measured in degrees positive from the equator.

*line_number
The alphanumeric flight-line identifier associated with the individual survey track-lines of a particular data file.

*longitude
This is the geographic longitude expressed as a decimal number measured positive to the EAST from the prime meridian. The units are degrees. The longitude range in the Minnesota region is -100 to -80, (indicating that all values are between 100 and 80 West longitude). Exceptions: all NURE data files (0420_???.bin), all Canadian Geological Survey data files (0500_???.bin), and two Minnesota Geological Survey data files (0280_012.bin and 0280_013.bin).

*longitude_west
This is the geographic longitude expressed as a decimal number measured positive to the WEST from the prime meridian. The units are degrees. The longitude range in the Minnesota region is 80 to 100 (indicating that all values are between 80 and 100 West longitude). NOTE, the only data files containing longitude parameters measured positive to the west are:
- all NURE data files (0420_???.bin),
- all Canadian Geological Survey data files (0500_???.bin),
- two Minnesota Geological Survey data files (0280_012.bin and 0280_013.bin).

*mag_anomaly
The magnetic anomaly computed with the 'Definitive' International Geomagnetic Reference Field (IGRF) value removed. (units of nano-Tesla).

*mag_anomaly_IGRF_removed_nT
The magnetic anomaly computed with the 'Definitive' International Geomagnetic Reference Field (IGRF) value removed. (units of nano-Tesla).

```
*mag_corr_flight_nT
Raw airborne magnetic measurement corrected by applying the 'BASE' magnetometer
value.
```

Example Menu Sections

Excerpts of the menu sections of the file GeoVu uses for the Global Vegetation Index (GVI) CD are shown below. This CD has two major types of data: bi-weekly and monthly GVI data. These two types of data are both in IDRISI-compatible files, but they were developed for two different versions of IDRISI, Version 3 and Version 4. (IDRISI is …). This means that the headers for the files have different formats. In addition to having different formats, the headers are in different directories than the data files.

```
*MAIN MENU
NULL
1) Bi-week GVI data         #*bi-week data
2) Monthly GVI data        #*monthly data
*monthly data
MAIN MENU
1) monthly GVI data of 1990  #*month of 1990
2) monthly GVI data of 1989  #*month of 1989
*month of 1990
monthly data
1) January          #\1month\cell\m9001.img
2) February           #\1month\cell\m9002.img
3) March             #\1month\cell\m9003.img
*month of 1989
monthly data
1) January          #\1month\cell\m8901.img
2) February           #\1month\cell\m8902.img
3) March             #\1month\cell\m8903.img
*bi-week data
MAIN MENU
1) Bi-week GVI data of 1991 #*bi-week of 1991
2) Bi-week GVI data of 1990 #*bi-week of 1990
*bi-week of 1991
```

```
bi-week data
1) week 2 of 1991 #\2biweek\cell\bw9102.img
2) week 4 of 1991 #\2biweek\cell\bw9104.img
3) week 6 of 1991 #\2biweek\cell\bw9106.img
4) week 8 of 1991 #\2biweek\cell\bw9108.img
5) week 10 of 1991 #\2biweek\cell\bw9110.img
*bi-week of 1990
bi-week data
1) week 2 of 1990 #\2biweek\cell\bw9002.img
2) week 4 of 1990 #\2biweek\cell\bw9004.img
3) week 6 of 1990 #\2biweek\cell\bw9006.img
4) week 8 of 1990 #\2biweek\cell\bw9008.img
5) week 10 of 1990 #\2biweek\cell\bw9010.img
```

Help Sections

Each menu item can (and ideally does) have an associated Help section with the form:

**submenu name*_help
help text

or

**filename*_help
help text

When a menu item is selected and the dialog's Facts button is clicked, text about the highlighted item is displayed. This is the information contained in a help section of the menu file. Usually one help section is written for each submenu item; sometimes it is also reasonable to write a help section for each file accessed through the menu system.

The title of a help section must be formed by appending **_help** to the sub-menu name or the base file name (without path or extension).

For example, the help section title must be **Special Vegetation Datasets_help** for the submenu specified:

```
Global Vegetation Datasets #*Special Vegetation Datasets
```

The help section title must be **gvdata_help** for the data file specified:

```
Special Vegetation Datasets #\special\data\gvdata.img
```

When two directories contain a file with the same base name, the same help section will appear for each unless the full path is used to create two different help section titles. The help section title must be **\special\january_data\gvdata_help** for the first data file specified; the section title must be **\special\february_data\gvdata_help** for the second.

```
Special Vegetation Datasets #\special\january_data\gvdata.img
Special Vegetation Datasets #\special\february_data\gvdata.img
```

Lastly, a help section can be formed from the right-hand text:

```
Special Vegetation Datasets_help
```

The lines of text can be in a separate file and referenced in the menu file. This is useful when an ASCII document or header file is already available. Use the syntax:

```
*submenu name_help
>&\directory\filename.ext
```

or

```
*filename_help
>&\directory\filename.ext
```

To cross-reference one help section to another within the menu file, use the syntax:

```
*submenu name_help
>*some_other_submenu_name_help
```

or

```
*filename_help
>*some_other_file_name_help
```

Equivalence Sections

You can include equivalence sections in a menu file instead of writing separate external equivalence files (**.eqv**). Equivalence sections are used for exactly the same purpose as equivalence files–to define GeoVu keywords, which correspond to file features, by equating data-specific variable names with the keywords.

Some features may be shared by all or most of the files in a data collection, others by subsets of files, and there may also be features that are unique to a particular file. By including equivalence sections in a menu file, you can accommodate both shared and unique features and eliminate the work of writing and maintaining external equivalence files.

The form of each equivalence section is as follows:

```
*title_eqv
contents of equivalence section
```

The syntax for the contents of the equivalence section is the same as the syntax for equivalence files. See chapter 3 for details.

Creating an Internal Equivalence Table

When GeoVu opens a new data file, it processes the full path and file name to find all relevant equivalence sections in the associated menu file. It then composes an internal equivalence table from this information. GeoVu finds the appropriate **_eqv** sections in the menu file by checking all or part of the data file path against the titles of all **_eqv** sections in the menu file for matches.

Equivalence Section Types and Precedence

GeoVu allows you to define the same keyword repeatedly and then uses rules of precedence to resolve the assignments taken from different equivalence sections. The order in which equivalence information is resolved depends on the type of equivalence section in which it was found. The three types of equivalence sections in order of precedence (from low to high) are default, general, and file. Equivalence information from a lower priority section can be overwritten by information from a higher priority section.

Default Equivalence Section

A menu file can have only one default equivalence section. Its title must be **default_eqv**. The default section is used to define keyword values (features) shared by the majority of files in the CD or data collection. Files that do not share these features can be covered by general or file equivalence sections as appropriate.

General Equivalence Sections

A menu file can have an unlimited number of general equivalence sections which define values for keywords shared by subsets of files. General equivalence sections are most often used to match directory paths and data type extensions.

The precedence of general equivalence sections relative to each other is determined by their order in the menu file–the further down in the file, the higher the precedence. The title of a general equivalence section is a character string preceded by `#` and trailed by `_eqv`, for example, `*#bin_eqv`. The character string can be two or more strings separated by `|` or `&`.

The `#` character and `_eqv` delimit the substring that is tested for matching against the full data file path. The match can be to a base file name, its extension, a partial path, a subdirectory, a portion of the directory hierarchy, and so on. For example, if the menu file associated with the DOS data file **d:\geovu\testdata\sample.bin** contains equivalence sections named:

```
*#bin_eqv
*#\testdata\sample_eqv
*#geovu\test_eqv
```

the equivalence information from all three sections will be included in the internal equivalence table according to precedence.

The `|` and `&` characters separating two or more strings in a section title are OR and AND operators. The OR operator (`|`) indicates that if either of the two strings is matched somewhere in the full data file path, the information in the **_eqv** section should be included. The `&` operator requires that both strings in the section title be found in the data file path.

File Equivalence Section

A file equivalence section defines features that are specific to a particular data file. Its title is the path name minus the CD or data drive name with the file extension replaced by **_eqv**, for example, `*\geovu\testdata\sample_eqv`

Example Equivalence Sections

The default equivalence section for the entire Global Vegetation Index CD:

```
*default_eqv
begin constant
     data_type char image
     image_type char bsq
     palette char GVI
     delimiter_item char \n
     _distance short 14
   header_file_ext char .doc
   ff_input_format char *idrisi_bfm
end constant
```

The equivalence section for files in the **1month** directory:

```
*#\1month_eqv
begin constant
   ff_header_format char *idrisi_header4_afm
   header_file_path char \1month\cdoc
end constant
begin name_equiv
     $file_title file%title
     $data_representation data%type
          char uchar char byte
          char short  char integer
          char float char real
     $number_of_rows rows
     $number_of_columns columns
     $minimum_value min. value
     $maximum_value max. value
     $data_value_unit data%units
     $right_map_x max.%X
     $left_map_x min.%X
     $upper_map_y max.%Y
     $lower_map_y min.%Y
     $map_projection ref.%system
     $grid_unit ref.%units
     $pixel_size unit%dist.
end name_equiv
```

The equivalence section for files in the **2biweek** directory:

```
*#\2biweek_eqv
begin constant
      header_file_path char \2biweek\bdoc
      ff_header_format char *idrisi_header3_afm
end constant
begin name_equiv
      $file_title image%title
      $data_representation data%type
            char uchar char byte
            char short  char integer
            char float char real
      $number_of_rows rows
      $number_of_columns columns
      $minimum_value minimum
      $maximum_value maximum
      $grid_size(x) cell%x
      $grid_size(y) cell%y
      $data_value_unit data%units
      $right_map_x max%X
      $left_map_x min%X
      $upper_map_y max%Y
      $lower_map_y min%Y
      $grid_unit ref.%units
      $comments comment
end name_equiv
```

Format Sections

A format description can be placed in a menu file instead of a separate **.bfm**, **.afm** or **.fmt** file. The format section title must be a name trailed by `_afm`, `_bfm`, or `_fmt`. These sections are linked to the data files by defining the applicable **ff_** keywords: **ff_input_format, ff_output_format, ff_format_fmt, ff_header_format.** The keywords are set to the format section titles in the equivalence table found in the menu file or an **.eqv** file.

Example Format Sections

EASEGRID
Format and directory-level equivalence sections from the EASEGRID menu:

```
*latlon_bfm
data 1 2 short 1

*#lon_eqv
begin constant
   ff_input_format char *latlon_bfm
   maximum_value short 180
    minimum_value short -180
end constant
```

Minnesota Aeromagnetics
Equivalence and format sections from the Minnesota Aeromagnetics CD menu file. Formats are described in format specification files and in sections of the CD menu file.

```
*default_eqv
begin constant
    data_type char point
    map_projection char lat/lon
    term_definition char minterms.txt
    histogram_dir char l">_"
end constant
*#\DATA\0101_eqv
begin constant
    ff_input_format char &\data\0101\minnf1.bfm
    ff_output_format char &\data\0101\minnf1.afm
end constant
*#\DATA\0102_eqv
begin constant
    ff_input_format char &\data\0102\minnf2.bfm
    ff_output_format char &\data\0102\minnf2.afm
end constant
*#\DATA\0103_eqv
begin constant
    ff_input_format char &\data\0103\minnf3.bfm
    ff_output_format char &\data\0103\minnf3.afm
end constant
*#\DATA\0104_eqv
begin constant
    ff_input_format char &\data\0104\mnusx.bfm
    ff_output_format char &\data\0104\mnusx.afm
end constant
```

```
*#\DATA\0105_eqv
begin constant
    ff_input_format char &\data\0105\mnusgs.bfm
    ff_output_format char &\data\0105\mnusgs.afm
end constant
*#\DATA\0201_eqv
begin constant
    ff_input_format char &\data\0201\minne1.bfm
    ff_output_format char &\data\0201\minne1.afm
end constant
*#\DATA\0202_eqv
begin constant
    ff_input_format char &\data\0202\minne2.bfm
    ff_output_format char &\data\0202\minne2.afm
end constant
*#\DATA\0203_eqv
begin constant
    ff_input_format char &\data\0203\minne3.bfm
    ff_output_format char &\data\0203\minne3.afm
end constant
*#\DATA\0204_eqv
begin constant
    ff_input_format char &\data\0204\minne4.bfm
    ff_output_format char &\data\0204\minne4.afm
end constant
*#\DATA\0205_eqv
begin constant
    ff_input_format char &\data\0205\nwwi.bfm
    ff_output_format char &\data\0205\nwwi.afm
end constant
*#\DATA\0206_eqv
begin constant
    ff_input_format char &\data\0206\nure1.bfm
    ff_output_format char &\data\0206\nure1.afm
end constant
*#\DATA\0207_eqv
begin constant
    ff_input_format char &\data\0207\cgs.bfm
    ff_output_format char &\data\0207\cgs.afm
end constant
*#\DATA\0208_eqv
begin constant
      histogram_dir char no_histogram_dir
      omit_grid_embedded_header char yes
      grid_origin char lowerleft
end constant
*\DATA\0208\Decimate_eqv
begin constant
    file_title char 0.426KM%Grid%of%Minnesota
      missing_flag float -99999.000
      maximum_value float 22416
      minimum_value float -7843
    ff_input_format char *decimate_afm
    input_repeat short 8
      output_repeat short 8
    ff_output_format char *decimate_bfm
    data_type char image
    left_map_x float -316.83960
    lower_map_y float 1157.6914
    upper_map_y float 1826.3616
    right_map_x float 272.03398
```

```
    grid_size(x) float 0.42671999
    grid_size(y) float 0.42671999
    number_of_rows float 1568
    number_of_columns float 1381
    number_of_bands short 1
    image_format char bsq
    map_projection char LCC
      central_meridian float -93.0
      base_latitude float 33.0
      fisrt_parallel float 33.0
      second_parallel float 45.0
end constant
*\DATA\0208\Glimpce_eqv
begin constant
    file_title char 0.40KM%Grid%of%Lake%Superior%residual%magnetics
    data_type char image
      missing_flag char 0.10000000E+31
      ff_input_format char *glimpce_afm
      input_repeat short 5
      output_repeat short 5
    left_map_x float -352.000001
    lower_map_y float -3.6000001
    upper_map_y float 457.6
    right_map_x float 351.6
    grid_size(x) float 0.4000000
    grid_size(y) float 0.4000000
    number_of_rows float 1154
    number_of_columns float 1760
    number_of_bands short 1
    image_format char bsq
    map_projection char LCC
      central_meridian float -88.5
      base_latitude float  46.0
      fisrt_parallel float 42.5
      second_parallel float 48.5
end constant
*\DATA\0208\North_eqv
begin constant
    file_title char 0.213KM%grid%of%Northern%Minnesota
    data_type char image
      missing_flag char -99999.000
    ff_input_format char *decimate_afm
    input_repeat short 8
    ff_output_format char *decimate_bfm
    output_repeat short 8
    left_map_x float -316.83960
    lower_map_y float 1434.8459
    upper_map_y float 1826.7882
    right_map_x float 272.46072
    grid_size(x) float 0.21336000
    grid_size(y) float 0.21336000
    number_of_rows float 1838
    number_of_columns float 2763
    number_of_bands short 1
    image_format char bsq
    map_projection char LCC
      central_meridian float -93.00
      base_latitude float 33.0
      fisrt_parallel float 33.0
      second_parallel float 45.0
end constant
```

```
*\DATA\0208\South_eqv
begin constant
    file_title char 0.213KM%grid%of%Southern%Minnesota
      missing_flag char -99999.000
    ff_input_format char *decimate_afm
      missing_flag char -99999.000
    input_repeat short 8
      output_repeat short 8
    ff_output_format char *decimate_bfm
    data_type char image
    left_map_x float -316.83960
    lower_map_y float 1157.6914
    upper_map_y float 1436.9796
    right_map_x float 163.00704
    grid_size(x) float 0.21336000
    grid_size(y) float 0.21336000
    number_of_rows float 1310
    number_of_columns float 2250
    number_of_bands short 1
    image_format char bsq
    map_projection char LCC
      central_meridian float -93.00
      base_latitude float 33.0
      fisrt_parallel float 33.0
      second_parallel float 45.0
end constant
*\DATA\0208\TCRESID_eqv
begin constant
    file_title char 0.213KM%grid%of%Twin%Cities%area
      missing_flag char -99999.0000
      ff_input_format char *tcresid_afm
    input_repeat short 6
      ff_output_format char *tcresid_bfm
      output_repeat short 6
    data_type char image
    left_map_x float 421.17264
    lower_map_y float 4932.0540
    upper_map_y float 5015.9045
    right_map_x float 535.96032
    grid_size(x) float 0.21336000
    grid_size(y) float 0.21336000
    number_of_rows float 394
    number_of_columns float 540
    number_of_bands short 1
    image_format char bsq
    map_projection char UTM
      central_meridian float -93.00
      base_latitude float 0
end constant
*decimate_afm
header 1 502 header 0
data   1 10 float 3
*decimate_bfm
data 1 4 float 3
*tcresid_afm
header 1 549 header 0
data 1 13 float 4
```

```
*tcresid_bfm
data 1 4 float 4
*glimpce_afm
header 1 563 header 0
data 1 16 char 0
```

IDRISI

IDRISI data and header format sections. Formats used for IDRISI binary data, the IDRISI4 ASCII header, and IDRISI3 ASCII header are shown as they appear in sections of the menu file **gvi.men**.

```
*idrisi_bfm
!.DOC%*idrisi_header_afm 1 9999 header 0
data 1 1 uchar 0
*idrisi_header4_afm
file%title  0 0 char 0
data%type   0 0 char 0
file%type   0 0 char 0
columns     0 0 short 0
rows        0 0 short 0
ref.%system 0 0 char 0
ref.%units  0 0 char 0
unit%dist.  0 0 double 7
min.%X      0 0 double 7
max.%X      0 0 double 7
min.%Y      0 0 double 7
max.%Y      0 0 double 7
pos'n%error 0 0 char 0
resolution  0 0 char 0
min.%value  0 0 short 0
max.%value  0 0 short 0
value%units 0 0 char 0
value%err   0 0 char 0
flag%value  0 0 char 0
flag%def'n  0 0 char 0
legend%cats 0 0 short 0
*idrisi_header3_afm
image%title 0 0 char 0
data%type   0 0 char 0
file%type   0 0 char 0
rows        0 0 short 0
columns     0 0 short 0
minimum     0 0 double 10
maximum     0 0 double 10
cell%x      0 0 double 10
cell%y      0 0 double 10
legend      0 0 short  0
backgrnd%%0 0 0 short 0
data%units  0 0 char 0
flag%value  0 0 double 10
min%X       0 0 double 10
max%X       0 0 double 10
min%Y       0 0 double 10
max%Y       0 0 double 10
ref.%system 0 0 char 0
ref.%units  0 0 char 0
comment     0 0 char 0
```

Header Section

This section is no longer used by GeoVu, but was used by the BETA and TEST1 releases.

5

Default Search Sequence

overview

Whatever …

This is the data format specification used for all EASEGRID time data files, available from the National Snow & Ice Data Center, from file **nsi_time.bfm**:

```
data 1 2 short 3
```

- ➔ GeoVu will expect to find and use the file **nsi_time.eqv** to get more information if the base file name of the data file is also **nsi_time** and both files reside in the same directory.
- ➔ Next, GeoVu will expect to find and use the format file **nsi_time.bfm** (or **.afm** or **.fmt**) to read the data if the base file name of the data is also **nsi_time** and both files reside in the same directory or if an **.eqv** file was found that sets up this association.
- ➔ If there is a menu (**.men**) file associated with the data set, GeoVu may find the reference to **nsi_time.eqv** or **nsi_time.bfm** there.
- ➔ If there is no menu file, the standard Format Type dialog is presented. If the file **stdform.nam** has **nsi_time** associated with a format type name, GeoVu will look for **nsi_time.eqv** - also **.bfm**, **.afm**, or **.fmt** - in the GeoVu working directory.

6

GeoVu Keywords

Keywords or reserved words are used to define properties that are meaningful to GeoVu. You can specify GeoVu's internal properties directly by assigning values to keywords in headers. GeoVu's internal properties can also be set to equivalent properties or keywords for each data source in equivalence files (**.eqv**) or equivalence sections (**_eqv**) in menu files (**.men**).
GeoVu can find the value of a keyword from (in order of precedence):

- the data file header if a keyword or its equivalent (as defined in an equivalence file or section) appears in the header format
- the constant part of an equivalence file or section
- **freeform.ini**, if you are running GeoVu under Windows; GeoVu first checks the GeoVu section, then the FREEFORM section
- keywords set as environment variables by using the **setenv** command in Unix
  Note! Keywords set at runtime are defined without a data type.

Functional Groups

keywords listed by functional group

.
.
.

Keyword Descriptions

The GeoVu keywords are listed alphabetically and a description is provided for each in the pages that follow. Any item enclosed in square brackets [ ] is optional. Because GeoVu looks for exact keywords, any character placed in front of a keyword creates a comment line, thus removing the keyword assignment.
Generally speaking, keyword values can be in any case. The three exceptions are the case-sensitive values for **omit_grid_embedded_header**, **preview**, and **user_select_data_type**. The value for **standard_format_name** is case-sensitive only if the operating system is.

auto_display

Description: Displays images and plots using a default search definition. Setting this keyword eliminates the Search/Create step from the process of creating a display. If this keyword is set and an image data set is selected, all data in the file is retrieved, which may take awhile for large images.
Type: yes, true, or no
Default: no
Required if you want to see displays automatically using all data.

band_#_unit

Description: The unit for band # in multi-band images. Specify the integer in place of #.
Type: char
Default: none
Required if you want to display the data value with the unit in the data value dialog.

base_latitude

Description: Defines base latitude for some map projections.
Type: numerical types
Default: none
Not required in GeoVu unless converting the grids to lat/lon value format.

bytes_per_pixel

Description: Defines how many bytes represent one pixel in an image.
Type: short
Values: 1, 2, 4, or 8
Required if the data representation is undefined.

category

Description: Defines the name for class #. Use the integer value in place of #. This keyword is from IDRISI GIS software and is the same as keyword **class_#**.
Type: char
Required if you want to show the class name in the data value dialog (same as **class_#**).

central_meridian

Description: Central meridian for some map projection. Usage is same as **base_latitude**.
Type: numerical types
Optional.

class_#

Description: Defines a class name and number (#). The name appears in the data value dialog.
Type: char
Required if you want to show the class name in the data value dialog.

data_byte_order

Description: Defines the data byte order.
Type: char
Values: big_endian, little_endian
Default: native byte order (data have the same order as the current machine)
Required if the data were produced on a different type of computer. We recommend strongly that this keyword be specified for all data collections. This makes it possible to view the collections on machines with native byte orders which are different than that on the machine where the data collection was created.

data_representation

Description: Defines how the data in the file represents the image.
Type: char
Values:

uchar - 1 bytes per pixel with possible data range from 0 to 255
short - 2 bytes per pixel with possible data range from -32768 to 32767
ushort - 2 bytes per pixel (unsigned short 0-65535)
long - 4-bytes-per-pixel integer
ulong - unsigned 4 bytes per pixel integer
float - 4-byte-per-pixel float
double - 8-byte-per-pixel float

Default: uchar

Required if the data representation is not uchar.

data_type

Description: Defines the spatial organization type of a data file.
Type: char
Values:
point – data normally displayed as points
vector – data normally displayed as points connected by lines
image/raster – data normally displayed as a 2-D grid
Default: No default value.
Required.

default_band

Description: Sets the default display band of a multi-band image.
Type: short
Values: 1- number_of_bands
Default: 1
Required if the default display band is not the first band.

delimiter_item

Description: A character that separates parameter name and value pairs in variable position headers. GeoVu can interpret a variable position header with lines having the following format:

```
parameter_name    delimiter_value   parameter_value   delimiter_item
```

The parameter name is an ASCII string which is used in GeoVu as the header variable name. For example: in **title : Soils Data Set \n**, "**:**" is the delimiter value between the parameter name and its value, and **\n** is the delimiter item.
Type: char
Values: Any character. For nonprinting characters such as newline, you can use standard C representation (\n).
Default: None
Required for variable position headers.

delimiter_value

Description: Defines the delimiter which separates parameter name from value in variable position header strings with the following format:

```
parameter_name    delimiter_value   parameter_value   delimiter_item
```

Type: char
Values: Any character. For nonprinting characters such as newline, you can use standard C representation (\n).
Default: None
Required if either **delimiter_item** or **_distance** is defined, but only for variable position headers.

_distance

Description: Defines the distance (number of bytes) between the first byte of `parameter_name` and the first byte of `parameter_value` in variable position header strings. This keyword has same role as the **delimiter_value**.
Type: short
Values: 1-32767 (integer).
Default: None
Requirement: Required for variable position headers.

dummy_embedded_header_length

Description: Defines embedded header length for data files which also have separate headers. If the relevant header information for a particular application is located in a separate header file, define this keyword and GeoVu will skip the irrelevant embedded header.
Type: numerical types
Default: None
Required if both a separate header file and an embedded header exist.

end_column

Description: Defines the end column number. Combining with **start_column**, GeoVu calculates the number of rows of an image.
Type: numerical types
Default: none
Either **number_of_columns** or both **start_column** and **end_column** must be defined.

end_row

Description: Combining with **start_row** to define the number of rows of an image.
Type: numerical types
Default: none
Either **number_of_rows** or both **start_row** and **end_row** must be defined.

false_east

Description: Defines the false east of some map projections; used only in map projection conversion. Written only to the header.
Type: numerical types
Requirement: Same as **base_latitude**.

ff_format_fmt

Description: Defines the location of a format description file (**.fmt**).
Type: char
Values: **section title* OR [&]*format file name*

- ➔ If the prefix is *, the format description is found in the menu file in the section *section title*.
- ➔ If the prefix is &, the format is located on the CD-ROM or data drive. In this case, you must include the directory path and file name relative to the CD or data drive in *format file name*.
- ➔ If there is no prefix, the format is defined in the file specified by *format file name* and located in the GeoVu working directory defined by GEOVUDIR.

Default: None
Optional.

ff_header_format

Description: Defines the location of a header format description.
Type: char
Values: **section title* OR [&]*format file name*

- ➔ If the prefix is *, the header format is found in the menu file in the section *section title*.
- ➔ If the prefix is &, the format is located on the CD-ROM or data drive. In this case, you must include the full directory path and file name in *format file name*.
- ➔ If there is no prefix, the format is defined in the file specified by *format file name* and located in the GeoVu working directory defined by GEOVUDIR.

Default: None
Optional. There are several ways to define a header format, including placing a reference to a header format file in the input data format specification, using the header format file name as the header variable name. See the *FreeForm User's Guide*.

ff_input_format

Description: Defines an input format specification file name.
Type: char
Values: **section title* OR [&]*format file name*

- ➔ If the prefix is *, the input format is found in the menu file in the section *section title*.

- ➔ If the prefix is &, the format is located on the CD-ROM or data drive. In this case, you must include the full directory path and file name in *format file name*.
- ➔ If there is no prefix, the format is defined in the file specified by *format file name* and located in the GeoVu working directory defined by GEOVUDIR.

Default: None
Optional.
Example taken from an equivalence table:

```
begin constant
    ff_input_format char &\sat\geo44\geo44.bfm
end constant
```

ff_output_format

Description: Defines the output format specification file.
Type: char
Values: **section title* OR [&]*format file name*

- ➔ If the prefix is *, the output format is found in the menu file in the section *section title*.
- ➔ If the prefix is &, the format is located on the CD-ROM or data drive. In this case, you must include the full directory path and file name in *format file name*.
- ➔ If there is no prefix, the format is defined in the file specified by *format file name* and located in the GeoVu working directory defined by GEOVUDIR.

Default: None
Optional.

file_title

Description: Defines the title which is used as the name of the data set and appears on the display window.
Type: char
Values: a character string which is the title for the data set
Default: same as the data file name
Optional.

first_parallel

Description: Defines the first parallel of some map projections. It is only used in the map projection conversion.
Type: numerical types
Not required in GeoVu unless converting the grids to lat/lon value format.(same as **base_latitude**).

format_dir

This is defined as a full path, used in GeoVu to locate format files. It must be defined (1) in a menu file, (2) in the GeoVu initialization file, or (3) as an environment variable.
Optional.

GEOVUDIR

This symbol must be defined as the full path of the GeoVu working directory, included in the GeoVu initialization file, or set as an environment variable. When GeoVu is installed under

Windows, the installation directory path is set to GEOVUDIR and placed in the initialization file **freeform.ini**.

grid_cell_registration

Description: Defines the relationship between the grid cell and the geographic location for the cell. It is only used in the map projection conversion.
Type: char
Values:

upperleft	The geo-location for the grid cell corresponds to the upper left corner of the cell.
lowerleft	The geo-location for the grid cell corresponds to the lower left corner of the cell.
lowerright	The geo-location for the grid cell corresponds to the lower right corner of the cell.
upperright	The geo-location for the grid cell corresponds to the upper right corner of the cell.
center	The geo-location for the grid cell corresponds to the center of the cell.

Default: center
Not required in GeoVu unless converting the grids to lat/lon value format (same as **base_latitude**).

grid_origin

Description: Defines the relationship between the map coordinate (georeferenced) and image coordinate (rows and columns) systems.
Type: char or short
Values:

upperleft	1	first point of the image is the upperleft corner of a map coordinate system and in column corresponds to a horizontal (x) map coordinate.
lowerleft	2	
lowerright	3	
upperright	4	
upperleft_y	5	first point of the image is the upperleft corner of a map coordinate system and in column corresponds to a vertical (y) map coordinate.
lowerleft_y	6	
lowerright_y	7	
upperright_y	8	

Default: upperleft
Required if the image is georeferenced and the grid origin is not the upperleft.

grid_size

Description: Defines, the pixel size of an image. If this keyword is defined, GeoVu assumes pixels are square.

Type: From uchar to double.
Default: 1
Required if the image is georeferenced.

grid_size(x)

Description: If the pixel is not square, this keyword defines the pixel size in x direction.
Type: from uchar to double
Default: 1
Required if the image is georeferenced and pixels are not square.

grid_size(y)

Description: If the pixel is not square, this keyword defines the pixel size in the y direction.
Type: from uchar to double
Default: 1
Required if the image is georeferenced and pixels are not square

grid_unit

Description: Specifies the unit for the pixel sizes.
Type: char
Values: character string
Default: None
Optional.

hdf_ref

Description: The HDF ref whose data will be displayed.
Type: numerical types
Default: the first data ref in an HDF file
Optional.

hdf_tag

Description: The HDF tag whose data will be displayed.
Type: numerical types
Default: the first data tag in an HDF file
Optional.

header_dir

Description: A full path, which GeoVu uses to locate header files. It must be defined in the menu file, in **freeform.ini** (Windows), or as an environment variable (Unix).
Optional.

header_file_ext

Description: Specifies the extension used for header files.
Type: char

Values: header file extension including the dot, e.g., .DOC. GeoVu assumes that the header file has the same base name as the data file but a different extension. GeoVu truncates the data file name extension and appends the string specified by this keyword.
Default: None
Requirement: Required if the header is in a separate file and **header_file_name** is undefined.

header_file_name

Description: Defines the header file name in which the data header is stored.
Type: char
Value: header file name, including the directory path, but without the drive letter if the file is on a CD. If the file is not a CD file, you can include a drive letter.
Default: None
Either **header_file_name** or **header_file_path** plus **header_file_ext** is required to access a separate header file.

header_file_path

Description: Defines the directory path for a separate header file. This keyword is combined with **header_file_ext** to form the complete name of a header file.
Type: char
Values: header file path (without drive letter if the file is located in a CD). If the separate header file is not in the same directory as the data file, this keyword redirects GeoVu to search the specified directory to find it.
Default: The same path and drive as the data file.
Required if **header_file_name** is undefined and the header file is located in a directory other than the one in which the data file resides.

header_length

Description: Defines the header length, i.e., number of bytes, for a file header of type header_embedded_varied (see **header_type**). For a separate header file, GeoVu uses the file size as the header length.
Type: short
Values: 1-32767 (short integer).
Default: None
Required if the header type is header_embedded_varied.

header_type

Description: Defines the header type associated with an input data file.
Type: char
Values:

- ➔ header_embedded–The header is embedded in the data file, and the location of each item (variable) is fixed. The total length of the header is also fixed.
- ➔ header_embedded_varied–The header is located at the head of the data file. The location of each item in the header and the total length of the header varies.
- ➔ header_separated–The header is in a separate file (other than the data file). The location of each item and the total length of the header file are fixed.
- ➔ header_separated_varied–The header is in a separate file. The item position, number of items, and file length varies.

Note! Variable headers can be ASCII or dBASE, but not binary.
Default: header_embedded
Optional.

histogram_dir

Description: Data collections produced by NGDC often have a set of pre-calculated histogram data for each variable. These histogram files are organized into one directory per data file. The name of this directory is specified (directly or indirectly) by **histogram_dir**.
Type: char
Values: character string

Default: None
Required if you have pre-calculated histograms in a data collection and want to let users access them.
Example: On the GRAVITY CD, each data file has its own directory, such as **\regional\anwr**. Each data file directory has a subdirectory called **lists**. GeoVu expects to find histogram files for the variables found in data file **\regional\anwr\anwr.bin** in the directory **\regional\anwr\lists**, so the keyword is defined as follows:

```
histogram_dir char lists\
```

Example: The Minnesota Aeromagnetics CD contains many data files in one directory but each file has its own subdirectory for histogram data. Files **0280_012.bin** and **0280_013.bin** are found in the path **\data\0101**. Under this path are the subdirectories L012 and L013, names formed by appending L to the part of the file name following the _ character. In this case, the keyword **histogram_dir** is defined as follows:

```
histogram_dir char L">_"
```

The character L is added to the front of the part of the data file name specified by the sting in quotes to form the name of the file's histogram directory. The first character in quotes is > or < followed by one or more characters. GeoVu searches the data file name to match the characters and takes the part of the file name to the right (if >) or left (if <) of the matched characters.

image_format

Description: The image format used to store multi-band images in a file. For single band image, all formats become the same.
Type: char
Values:
➔ BSQ–Band Sequential. In the file, image band 1 is stored first, then band 2, then band 3, etc.
➔ BIL–Band Interleaved by Line. The storing sequence is: first rows of band 1 , first rows of band
2, ..., first rows of band n, and second rows of band 1, second rows of band 2, ..., second rows of
band n, etc.
➔ BIP–Band Interleaved by Pixel. The storing sequence is: first pixel of band 1, first pixel of band
2,..., first pixel of band n, second pixels of band 1, second pixel of band 2, ..., second pixel of
band 2, etc.
➔ BIP2–Band Interleaved by 2 Pixels. The storing format is: first 2 pixels of band 1, first 2 pixels
of band 2, ..., first 2 pixels of band n, etc.
Default: BSQ
Required if the format is not BSQ.

index_file_name

Description: The index file created by the FreeForm **sortpoint** utility is usually located in the same directory as the data file. If the index file is not in that directory or you want to use index file in other directory, this keyword is used.
Type: char
Values: file name with path
Default: none
Required if you want to redirect the index file.

input_repeat

Description: If input data is ASCII, GeoVu assumes that after each record there is an end-of-line character. If input lines actually consist of more than one record, this keyword tells GeoVu the number of records per line. Also use for image files with multiple pixels per line.
Type: short
Values: 1-32767
Default: 1
Required for ASCII data files with more than one record per line or image files with more than one pixel per line.

keep_preview

Description: Keeps a preview image open after it has been displayed.
Type: char
Value: yes, keep the preview image
no, delete the preview image and close the preview data file
Default: yes
Optional.

latitude_max

Description: Same as **upper_map_y**. Special name for lat/lon projection.

latitude_min

Description: Same as **lower_map_y**. Special name for lat/lon projection.

left_map_x

Description: The left x (map coordinate) boundary of the image.
Type: float
Default: None
Required if the image is georeferenced (same as **right_map_x**).

longitude_max

Description: Same as **right_map_y**. Special name for lat/lon projection.

longitude_min

Description: Same as **left_map_y**. Special name for lat/lon projection.

lower_map_y

Description: The lower y (map coordinate) boundary of the image.
Type: float
Default: None
Required if the image is georeferenced (same as **right_map_x**).

map_projection

Description: Defines the name of the map projection of an input data file. If the data type is IMAGE and map projection is lat/lon, GeoVu supports overlays of coastlines, political boundaries, and location markers. When the projection is lat/lon, you must define other keywords, e.g., **right_map_x**, **left_map_x**, **upper_map_y**, and **lower_map_y**.
Type: either char or short. If the type is short, the range is from 0 to 23. The values 0-22 correspond to the projection names listed above while 23 means "other projections" (not listed).
Values: for the character type, the values are:

none
UTM
State plane
Albers conical equal area
Lambert conformal conic
Mercator
Polar steterographic
Polyconic
Equidistant conic
Transverse Mercator
Steterographic
Lambert azimuthal equal area
Azimuthal equidistant
Gnomonic
Orthographic
General vertical near_side perspective
Sinusoidal
Equirectangular
Miller cylindrical
Van der Grinten
Oblique Mercator
Lat/Lon

Default: "none" (not projected)
Optional.

maximum_value

Description: The maximum value in the data file. If a **missing_flag** value is defined, you should not include this value. This value is used in conjunction with **minimum_value** to scale data (normally image data) for display purposes. If the file has multiple variables

(multiple bands), you should define each variable's maximum value by defining the keyword **var*_band#_*max**.
Type: from uchar to double
Values: Maximum value in the data file.
Default: Maximum value for the data type
Required

minimum_value

Description: The minimum value in the input data file. If a **missing_flag** value is defined, you should not include this value. This value is used in conjunction with **maximum_value** to scale data (normally image data) for display purposes. If the file has multiple variables (multiple bands), you should define each variable's minimum value by defining the keyword **var*_band#_*min**.
Type: from uchar to double, minimum data value in the file
Default: 0
Required

missing_flag

Description: For a gridded image file, some pixels may not have a meaningful value. Those pixels are filled with the flag value defined by this keyword. GeoVu automatically displays missing pixels using the value 0.
Type: from uchar to double
Default: None
Optional.

number_of_bands

Description: Defines how many images with same size are stored in the file.
Type: short
Values: 1-32767, number of bands
Default: 1
Requirement: Required if the number of bands is more than 2. If the **image_format** is BIP, GeoVu uses the number of variables in the input format as this value.

number_of_classes

Description: Defines the number of classes in an image. If this keyword is defined, GeoVu uses it to set the **minimum_value** = 0 and **maximum_value = number_of_classes**-1.
Type: short
Values: 1-32767
Default: None
Required if **minimum_value, maximum_value, var*_band#*_max** and **var*_band#*_min** are undefined.

number_of_columns

Description: Defines the number of columns for an image (grid) file.
Type: short
Values: 1-32767
Meanings: number of columns in the image
Default: None
Required if the data type is image

number_of_display_colors

Description: GeoVu's default number of display colors is 16. If more than 16 colors is possible (256 or above), the 16-color display allows you to display more than one window with the correct colors. However, if higher spectral resolution is required, you can set this value to a larger number (less than or equal to the number of system colors and not larger than 256).
Type: short
Values: >= 16 && <= 256
Default: 16
Optional.

number_of_rows

Description: Number of rows for an image (grid) file.
Type: short
Values: 1-32767, number of rows in the image
Default: None
Required if the data type is image

omit_grid_embedded_header

Description: When using GeoVu to convert a data file (the view type is Write To Disk), the embedded header of the original file is copied to the disk file without changes. Set this keyword to Yes to suppress this header in the new disk file.
Type: char
Values: Yes, No
Default: No
Optional.

output_repeat

Description: GeoVu puts end-of-line characters at the end of each record that it writes out to a new ASCII disk file. (The files are written one record per line.) To collect more than one record per line, define this keyword with a value.
Type: short
Values: 1 - 32767
Default: 1
Required if you want to put multiple records into one line.

PAL_FILE (a control in **freeform.ini**, not a GeoVu keyword)

Description: The name of the list file containing names of palettes to be used by GeoVu at start-up. If this is not set in **freeform.ini**, the default file used by and distributed with GeoVu is **geopal.lst**. This file should be in the working directory defined by GEOVUDIR.
Optional.

palette

Description: The name of the palette used by GeoVu for image displays. Use this keyword only for image data collections. Note that GeoVu reads **.pal** files only if the name has been linked into the palettes list file.
Type: char
Values: Names listed in the palettes list file plus a 16-shade Black/White and a 16-color VGA palette.
Default: the 16-shade greyscale palette called Black/White
Optional.

plot_drawing_default

Description: Changes from default plotting with points to plotting with connected lines.
Type: char

Values: connected_line is the only possible value; omit it and the GeoVu default for plotting, points, is used.
Default: none
Optional.
Example (emphasis added):

```
*default_eqv
begin constant
      data_type char point
      .
      .
      .
      plot_drawing_default char connected_lines
      plot_x_default char month
      plot_y_default char year
end constant
```

plot_x_default

Description: User-defined value that appears as a default choice on the lists in plot display windows. These lists show independent and dependent (x, y) variable selections.
Type: char
Values: Name of variable exactly as it appears in the data set.
Default: latitude, longitude if found
Optional. Use only for point data and only if GeoVu-supplied defaults are inappropriate.
Example (emphasis added):

```
*default_eqv
begin constant
      data_type char point
      .
      .
      .
      plot_drawing_default char connected_lines
      plot_x_default char month
      plot_y_default char year
end constant
```

plot_y_default

Description: User-defined value that appears as a default choice on the lists in plot display windows. These lists show independent and dependent (x, y) variable selections.
Type: char
Values: Name of variable exactly as it appears in the data set.
Default: latitude, longitude if found
Optional. Use only for point data and only if GeoVu-supplied defaults are inappropriate.
Example (emphasis added):

```
*default_eqv
begin constant
      data_type char point
      .
      .
      .
      plot_drawing_default char connected_lines
      plot_x_default char month
      plot_y_default char year
end constant
```

preview

Description: Turns on display of preview images.
Type: char
Values: Yes or No
Default: none
Required if preview images are to be displayed.

preview_command

Description: Displays the preview file using an alternative command. The syntax is the same as for **slide_show_command**.
Type: char
Default: none
Required if you want to use an alternative preview command.

preview_ext

Description: Enables GeoVu to use an alternative file for preview displays.
Type: char
Values: the extension of the alternative file name
Default: none
Required if you want the preview file to be the alternative file.

preview_path

Description: Defines the path of the alternative preview file.
Type: char
Values: the path of the alternative file
Default: none
Required if the alternative preview file is not in the same directory as the data file.

record_count

Description: Gives the number of data points per record in data sets which include record headers followed by a fixed number of data points. This defines the number of records.
Type: numerical types
Required for record type data.

record_header_length

Description: Number of bytes for the record header.
Type: numerical types
Required for record headers.

right_map_x

Description: The right x (map coordinate) boundary of the image.
Type: float
Default: None
Required if the image is georeferenced.

row_number of_bytes

Description: Defines number of bytes for each image row.
Type: numerical types $\leq$ long
Default: None
Required if keyword **user_define_row_structure** is defined.

row_start_position

Description: Defines the start byte number for the first element of image in a file.
Type: numerical types $\leq$ long
Default: None
Required if keyword **user_define_row_structure** is defined.

row_step

Description: Defines number bytes between rows.
Type: numerical types $\leq$ long
Default: None
Required if keyword **user_define_row_structure**.

scaling

Description: Defines the scaling type used in the conversions of real data into the image for display.
Type: char
Values: linearzero: linearly compress the data into the range defined by the keyword **number_of_display_colors**.
Frequency: Not implemented in this release.
User defined: Not implemented in this release.

Histoeq: First linearly compress the data into 256 data classes, then equalize the 256 classes into the range defined by **number_of_display_colors**.
Default: Linear
Optional.

second_parallel

Description: Defines the second parallel for some map projection.
Type: numerical types
Optional.

slide_show_command

Description: Defines the command used to run the slide show, for example, **slide_show_command char pcx@d:\views\wingif.exe%$*filename***. This command indicates that the slide show will use a program called **wingif.exe** in the **d:\views** directory, and the command accepts a file name argument. The **%** character represents a white space, and the **$** sign tells GeoVu to replace ***filename*** with the actual file name. The characters **pcx** before the @ sign tell GeoVu to invoke this command when the slide show file is a pcx file. This keyword works only on the Microsoft Windows version of GeoVu.
Type: char
Required if you want to use program other than GeoVu to display the slide show files.

slide_show_ext

Description: Defines the slide show file type. On some CD-ROMs, some or all data files are accompanied by alternative slide show files. These files are usually small so they can be displayed quickly. This keyword directs the slide show to display those files instead of the data files. If this keyword is defined, GeoVu will ask the user which files to display when the slide show is invoked.
Type: char
Values: the extension of the alternative slide show file names
Required if there are alternative slide show files.

slide_show_path

Description: Defines the path for the alternative slide show file.
Type: char
Values: the path of the alternative slide show file
Required if the alternative slide show file is not in the same directory as the data file.

standard_format_name

Description: If the file format is a standard format recognized by GeoVu, you define this keyword and GeoVu automatically gets all information required for displaying the image. See **stdform.nam** in GEOVUDIR for a list of standard formats.
Type: char
Values:
Values are case-sensitive if the operating system is case-sensitive.

erdas_ra	ERDAS raster
idrisi3	IDRISI raster v3; former value was idrisi_r
idrisi4	IDRISI raster v4
geovu_ra	GEOVU raster
user	Binary raster user defined
pcx	PCX
mcidas_d	McIDAS PC IMAGE
mcidas_u	McIDAS UNIX IMAGE
gac16	GAC unpacked level 1b data

Default: none

start_column

Description: The start column of an image. Combining with **end_column** to calculate the number of columns of an image.
Type: numerical types
Either **number_of_columns** or both **start_column** and **end_column** must be defined.

start_row

Description: The start row of an image. Combining with **end_row** to calculate the number of rows of an image.
Type: numerical types
Either **number_of_columns** or both **start_row** and **end_row** must be defined.

term_definition

Description: Gives the name of the text file which contains the definitions for variable names (a glossary). If this keyword is defined, the user can view the definitions of a variable when changing XY-Plot variables in the XY-Plot dialog.
Type: char
Values: [&]*filename*
➔ If prefixed with &, the definition file is located on a CD-ROM. In this case, the file name must
include the whole path but without the CD-ROM drive letter.
➔ If there is no prefix &, the file is located in the GeoVu working directory. In this case, the file
name must not include the file path.

upper_map_y

Description: The upper y (map coordinate) boundary of the image.
Type: float
Default: None
Optional.

UTM_x_max

Description: Same as **right_map_y**. Special name for UTM projection.

UTM_x_min

Description: Same as **left_map_y**. Special name for UTM projection.

UTM_y_max

Description: Same as **upper_map_y**. Special name for UTM projection.

UTM_y_min

Description: Same as **lower_map_y**. Special name for UTM projection.

UTM_zone

Description: Defines the UTM zone. It is only used in the map projection conversion.

Type: numerical types
Default:
Required if the projection is UTM.

user_define_row structure

Description: Row structure is the basic data structure for GeoVu to access grid data. If the keyword is defined, GeoVu looks for the keywords **row_start_position**, **row_number_of_bytes**, and **row_step** for row structure information.
Type: char
Values: any
Default: None
Required if you want to define the row structure for GeoVu to access.

user_select_data_type

Description: This keyword is only used for HDF files. If this keyword is defined, GeoVu looks for the keyword **data_type** to determine how to display data in the HDF file. If this keyword is not defined, GeoVu displays a scientific data set as an image and a vdata as point data.
Type: char
Values: Yes or No
Default: No

value_type

Description: Affects the conversion of data into the graphics image buffer. When this value is defined, GeoVu will not linearly scale the data value when the number of classes is less than the number of display colors, so that each class has its own unique color.
Type: char
Values: class_data
Default: numerical_data
Use this keyword if you don't want GeoVu to scale the data and the **number_of_classes** is less than the **number_of_display_colors**.

value_unit

Description: Defines the unit name for the data values in an image if the image has only one band. The unit appears in the data value dialog.
Type: char
Required if you want the unit name appeared on the data value dialog.

var_*band#*_max

Description: The maximum value for band #. Use the integer value in place of *band#*.
Type: uchar to double
Default: None
Optional. If the image file has multiple bands and this keyword is not defined, GeoVu uses **maximum_value**.
Example: The maximum value for band 2 is 300.0.

```
var_2_max float 300.0
```

*var_band#*_min

Description: The minimum value for band #. Use the integer value in place of *band#*.
Type: uchar to double
Default: None
Optional. If the image file has multiple bands and this keyword is not defined, GeoVu uses **minimum_value**.
Example: The maximum value for band 2 is 300.0.

```
var_2_max float 300.0
```

*var_name*_max

Description: This keyword is the same as the **var_*band#*_max** but you can use the variable name directly in the keyword.
Type: char
Optional. If the image file has multiple bands and this keyword is not defined, GeoVu uses **maximum_value** (same as **var_*band#*_max**).

*var_name*_min

Description: This keyword is the same as the **var_*band#*_min** but you can use the variable name directly in the keyword.
Type: char
Optional. If the image file has multiple bands and this keyword is not defined, GeoVu uses **minimum_value** (same as **var_*band#*_min**).

*var*_name_unit

Description: Defines the unit for a variable. Place the actual variable name in place of *var_name*. This keyword is the same as the **band_#_unit**.
Type: char
Required if you want to display the data value with the unit in the data value dialog (same as for **band_#_unit**).

view_command

Description: Displays a data file using an alternative command. The format is the same as for **slide_show_command**.
Type: char
Default: none
Required if you want to use an alternative to GeoVu's default view command.

view_ext

Description: Enables GeoVu to use an alternative file for display.
Type: char
Values: the extension of the alternative display file name
Default: none
Required if you want to display an alternative file.

7

Menu File Utilities

There is currrently one menu file utility, **menutil**, which can provide major diagnostic assistance as you write GeoVu menu files. It is used to check menu files for proper syntax and completeness. It also constructs menu file outlines and adds indexing into menu files.

Checking Menu Files

Every time **menutil** runs, it checks for and removes spaces at the end of lines regardless of the arguments you specify. The only required argument for **menutil** is the name of the input menu file. All other arguments are optional and can be in any order following the input file name. The arguments are used for several different purposes including specifying input and output menu files, initiating and defining the scope of the check, and creating menu indexes and outlines.

The **menutil** command line has the following form:

menutil menu_file [-d CD_DRIVE] [-n new_menu_file] [-check] [-nodata] [-nohelp] [-fill] [-fixhelp]
[-trim] [-index] [-trimindex] [-tree] [-lower]

Note! To see a summary of command line usage for **menutil**, just enter the program's name on the command line without any arguments.

menu_file

Name of the menu file to check and make changes to (such as removing spaces at the end of lines), if necessary. The expected extension for a GeoVu menu file is **.men**.

-n new_menu_file

Option flag followed by the name of the file to store the modified menu in. The input menu file name specified by **menu_file** is used by default. If you specify a new name, the original menu file retains the name **menu_file.men**. If you do not specify a new name, the original file is stored in **menu_file.old**.

Warning! Do not run **menutil** on a file with the **.old** extension.

-d CD_DRiVE

Option flag followed by the drive letter (PC) or path (Unix) where the files (data, help, etc.) mentioned in the menu file are located (wheteher on CD-ROM, hard drive, or diskette). Default is the current drive or path.

-check

Option flag that tells **menutil** to check the menu file for syntax errors (limited to checking for spaces at the end of lines) and omissions. It looks for the following menu sections:

```
MENU_NAME
VOLUME_ID
SLIDE SHOW
ANIMATION
INTRODUCTION
```

These sections are not all required in every menu file, but **menutil** lets you know if they are not there.

In addition, **menutil** checks for a help section for each menu item and for referenced menu sections, where the reference is the menu name preceded by `#*`. It also checks to see that all the files mentioned in the menu file exist.

Note! The check for spaces at the end of lines is performed every time **menutil** runs whether or not you use the **-check** option.

-nodata

Suppresses the check for data files when used with the **check** option.

-nohelp

Suppresses the check for help files (denoted by `>&`) when used with the **check** option.

-fill

Option flag that causes **menutil** to find all references to help files in the menu file (of the form `>&help_file`) and replace the references with the text of the referenced help files.

-index

Creates a Menu Index section at the top of the menu file providing information about the location of menu sections. The menu index speeds up GeoVu's display of data. The Menu Index section is of the form:

```
*MENU INDEX
   index entry
   index entry
      .
      .
      .
END_MENU_INDEX
```

Each entry in the index indicates a section's offset from the top of the menu file in bytes and number of lines. The menu index occupies many lines if the menu file has numerous sections. If you make any changes to the menu file, you should reindex it, although **menutil** attempts to reindex automatically if it finds a corrupt index.

-trimindex

Removes the Menu Index section from the menu if one exists.

-tree

Creates three files showing menu file structure at different levels of detail.

The file **menutree** is a schematic tree in outline form of the menu and menu item sections in the menu file.

Example

```
.  Impact
.  .  Engineering aspects
.  .  .  Performance of engineered structures
.  .  .  .  Industrial facilities
.  .  .  .  .  Destroyed equipment
.  .  .  .  .  Spitak flour mill
                  .
                  .
                  .
```

The file **filetree** adds to the tree the files associated with each menu item.

Example

```
.  Impact
.  .  Engineering aspects
.  .  .  Performance of engineered structures
.  .  .  .  -\spitakdb\colcomme\sc00101c.pcx
.  .  .  .  -\spitakdb\colslab\sc00101c.pcx
.  .  .  .  Industrial facilities
.  .  .  .  .  -\spitakdb\colindus\sc00101c.pcx
.  .  .  .  .  -\spitakdb\colindus\sc00102c.pcx
.  .  .  .  .  Destroyed equipment
.  .  .  .  .  .  -\spitakdb\colindus\sc00103c.pcx
.  .  .  .  .  .  -\spitakdb\colindus\sc00103d.pcx
.  .  .  .  .  Spitak flour mill
.  .  .  .  .  .  -\spitakdb\colindus\sc00104c.pcx
.  .  .  .  .  .  -\spitakdb\colindus\sc00104d.pcx
                  .
                  .
                  .
```

The file **filelist** shows all the lines in the menu file having the form

```
menu item name #\directory\filename
```

That is, each line shows a menu item name as it appears in the user interface followed by the name of the file the menu item invokes.

Example

```
Thrust fault #\ngdcpcx\64701119.pcx
Right-lateral Strike Slip Fault near Spitak #\ngdcpcx\64701120.pcx
Hydrodynamic precursors #\spitakdb\colpredi\sc00102c.pcx
Deformation precursors #\spitakdb\colpredi\sc00103c.pcx
Electro-magnetic precursors #\spitakdb\colpredi\sc00104c.pcx
Biological precursors #\spitakdb\colpredi\sc00105c.pcx
      .
      .
      .
```

-fixhelp

Changes incorrectly assigned help section titles. Recall that the title of a help section is formed by appending `_help` to the submenu name (preceded by `#*`) or the base file name (preceded by `#\directory\`). If you have specified the following menu item:

```
Global Vegetation Datasets #*Special Vegetation Datasets
```

and have based the corresponding help section title on the user interface name for the menu item, i.e., `Global Vegetation Datasets_help` , **menutil** changes it to `Special Vegetation Datasets_help`. Be aware that **menutil** cannot detect all incorrectly assigned help titles of this sort.

-lower

Changes all file names in the menu file to lowercase.

Appendix A

Example Menu File

GRAVITY CD

```
*VOLUME_ID
GRAV_91A
*default_eqv
begin constant
    data_type char point
    term_definition char terms.txt
      histogram_dir char lists\
      map_projection char lat/lon
end constant
*#grids_eqv
begin constant
      histogram_dir char no_histogram_dir
end constant
*INTRODUCTION
                                  Gravity CD-ROM
                                   Data Retrieval
                              (Alpha Release, March 1992)

                        NOAA's National Geophysical Data Center
                      325 Broadway, Boulder, Colorado 80303 USA
                                  FAX: 303-497-6513

                    For support contact:
                    Allen M. Hittelman (303-497-6591)
```

```
*MAIN MENU
NULL
 US Station Data (and Source Documentation)  #*US Station Data and Source
Documentation
 Regional Surveys     #*Regional Surveys
 Gravity Networks     #*Gravity Networks
 Absolute Gravity     #\absolute\ngsagrav.dat
 Satellite Measurements #*Satellite Measurements
 Gravity Anomaly Grids  #*Gravity Anomaly Grids
*US Station Data and Source Documentation
MAIN MENU
 Defense Mapping Agency (1991 ed.) -- 910,858 observations  #\regional\dma\dma.bin
 NOAA National Geodetic Survey (1990 ed.) -- 1,496,805 obs. #\regional\ngs\ngs.bin
 DMA Author Index File #\regional\dma\91dmauth.dat
 NGS Documentation File  #\regional\ngs\ngsgrav.doc
*NGS Base
Gravity Networks
 Gravity Network Base Station Descriptions #\regional\statdesc\stadesc1.dat
 Station Name Sorted Alphabetically      #\regional\statdesc\stadesc2.dat
 Station Names Sorted by State         #\regional\statdesc\stadesc3.dat
*Regional Surveys
MAIN MENU
 Alaska  #*Alaska
 Calif/Nevada   #*Calif/Nevada
 Indiana        #\regional\indiana\indiana.bin
 New Mexico     #\regional\newmex\newmex.bin
 Oregon         #\regional\portland\portland.bin
 Wisconsin      #*Wisconsin
 Utah/Colorado  #\regional\vernal\vernal.bin
 Africa         #\regional\africa\africa.bin
 Andes          #\regional\andes\andes.bin
 Antarctic      #\regional\antarc\antarc.bin
 Egypt          #\regional\egypt\egypt.bin
 Japan          #\regional\jodc\jodc.bin
 South America  #\regional\samerica\samerica.bin
*Alaska
Regional Surveys
 Arctic National Wildlife Reserve  #\regional\anwr\anwr.bin
 National Petroleum Reserve - Alaska    #\regional\npra\npra.bin
 Holitna Area (with topographic elevation model) #\regional\holitna1\holitna1.bin
 Holitna Area (with altimeter elevation model)   #\regional\holitna2\holitna2.bin
*Calif/Nevada
Regional Surveys
 California (and Southern Nevada) #\regional\calif\calif.bin
 Cadiz Area #\regional\cadiz\cadiz.bin
 Nevada     #\regional\nevada\nevada.bin
*Wisconsin
Regional Survey
 Ashland Area (WISC1)         #\regional\wisc1\wisc1.bin
 Rhinelander Area (WISC2A)       #\regional\wisc2a\wisc2a.bin
 Prentice Area (WISC2B)       #\regional\wisc2b\wisc2b.bin
*Gravity Networks
MAIN MENU
 National Geodetic Survey US Monument Network  #\regional\90ngsnet\90ngsnet.bin
 National Geodetic Survey Gravity Base Station Description #*NGS Base
 Defense Mapping Agency Global Network           #\regional\dmanet\dmanet.bin
*Satellite Measurements
MAIN MENU
 GEOS3/SEASAT data  #*GEOS3
 GEOSAT - Ascending Orbital Data (from 44 repeat cycles)  #\sat\geosat\geo44asc.bin
 GEOSAT - Descending Orbital Data (from 44 repeat cycles) #\sat\geosat\geo44des.bin
```

```
*Gravity Anomaly Grids
MAIN MENU
 6-km       grid of N. America (DNAG)  #\grids\dnaggrav.bin
 2.5-min    grid of N. America         #\grids\dgrav.bin
 4-km       grid of US (SEG)         #\grids\seggrav.bin
 2.5-min    isostatic grid of US     #\grids\isostat.bin
 4-km       isostatic residual grid of US   #\grids\isores.bin
 8-km       isostatic gravity grid of US    #\grids\isoreg.bin
 8-km       isostatic topo grid of US       #\grids\isotop.bin
 10-min     geoid grid of Canada              #\grids\canada10.bin
 .1*.05 deg grid for southern ocean (60-72 S. Lat)  #\grids\sograv\sograv.bin
 30-min     global grid                             #\grids\rapp30m\rapp30m.bin
 1-deg      global grid                             #\grids\rapp1deg\rapp1deg.bin
*GEOS3
Satellite Measurements
         0  72    0  44.875      #\sat\geos3\geos1.bin
         0  72   45  89.875      #\sat\geos3\geos2.bin
         0  72   90 134.875      #\sat\geos3\geos3.bin
         0  72  135 179.875      #\sat\geos3\geos4.bin
         0  72  180 224.875      #\sat\geos3\geos5.bin
         0  72  225 269.875      #\sat\geos3\geos6.bin
         0  72  270 314.875      #\sat\geos3\geos7.bin
         0  72  315 359.875      #\sat\geos3\geos8.bin
  -71.875   0    0  44.875      #\sat\geos3\geos9.bin
  -71.875   0   45  89.875      #\sat\geos3\geos10.bin
  -71.875   0   90 134.875      #\sat\geos3\geos11.bin
  -71.875   0  135 179.875      #\sat\geos3\geos12.bin
  -71.875   0  180 224.875      #\sat\geos3\geos13.bin
  -71.875   0  225 269.875      #\sat\geos3\geos14.bin
  -71.875   0  270 314.875      #\sat\geos3\geos15.bin
  -71.875   0  315 359.875      #\sat\geos3\geos16.bin
*U.S. Station Data and Source Documentation_help

The primary data sets on this compact disc are the station data
compiled by the Defense Mapping Agency (DMA) and NOAA's National
Geodetic Survey (NGS).  Shortly before pressing this compact
disc, we received a 1991 update to the DMA data, and these
updates were included.

A key difference between these two files is that DMA data do not
include any terrain data, while the NGS data are terrain
corrected (in locations for which this correction is deemed
necessary).

The DMA Author Index File is a form of documentation used in both
of these data sets.  It attempts to document within each record
(in the "source_id" field) information about the origins of the
survey. The level of documentation is minimal and often subject
to interpretation.
```

The NGS Vertical Control Station Description is a file of "Bench Mark Descriptions" for gravity bases. The file is viewable as a text, and additional files help users identify stations by alphabetical sorts which present the information by station name or by state name.

*Regional Surveys_help

Many regional data sets have been contributed to the National Geophysical Data Center over the years. These data sets range, geographically, from several miles to continental in scale. Some selections will result in another regional selection screen. There are four sub-regions in Alaska, and three sub-regions in both "Calif/Nevada" and "Wisconsin".

After selecting a unique data set, an "Area of Interest" screen will appear. From this "Area of Interest" screen, three special browse and documentation features exist: (1) To obtain additional documentation on the data set (i.e., data contributor, processing parameters, etc.) press the help key -- "?" or any function key. (2) To view a histogram distribution plot of any field in the data set press "H", select a field, and then press ENTER; a histogram will appear. (3) To view a listing of the distribution, on the last histogram screen, press "L".

*Gravity Networks_help

Two gravity networks are available on the CD-ROM.

The National Geodetic Survey (NGS) network is a special subset of the NGS station data base for stations designated as monuments. Data are exclusively in the conterminous United States.

The Defense Mapping Agency Global Network is international in scope.

After selecting one of these network data sets, an "Area of Interest" screen will appear. From this "Area of Interest" screen, three special browse and documentation features exist: (1) To obtain additional documentation on the data set (i.e., data contributor, processing parameters, etc.) press the help key -- "?" or any function key. (2) To view a histogram distribution plot of any field in the data set press "H", select a field, and then press ENTER; a histogram will appear. (3) To view a listing of the distribution, on the last histogram screen, press "L".

*Absolute Gravity_help

The National Geodetic Survey's absolute gravity data values are presented as a scrollable tabular text file. The NGS Absolute Gravity data (46 stations) were received in May 1990. Principal gravity parameters include Gravity Value, Uncertainty, and Vertical Gradient. The data are randomly spaced within the boundaries of the US, Bermuda, Hawaii, and Canada.

```
*Satellite Measurements_help

There are two satellite-derived data sets available:

(1) GEOS3/Seasat altimeter data were used to compute gravity
anomalies and sea surface heights on a 0.125 degree grid by Dr.
Richard H. Rapp.

(2) GEOSAT altimeter data, from 44 repeat cycles, were used by
Dr. David T. Sandwell to compute geoid and gravity anomaly
profiles. Two unique data sets exist for ascending and descending
orbits. These data have internal headers, and a "C" program
exists on the CD-ROM to read these headers.



*Gravity Anomaly Grids_help


Eleven gravity grids are included in this CD-ROM.  Many of these
data sets are also present on the Geophysics of North America
CD-ROM (published by the National Geophysical Data Center in
1989).
To select a grid,  press the number or letter associated with
that menu option, followed by pressing the ENTER key.

Area searches are permissible only after a unique data set is
selected.  Since each data set has its own unique format and
projection, it was decided to limit area selections in this
manner. For example, a user cannot simply ask for all data in a
specified area.

After selecting a unique data set, the user will be presented
with an "Area of Interest" screen.  This screen will:
    - identify the specific file name on the CD-ROM,
    - present the file sizes (in ASCII or binary),
    - display the spatial limits and attributes of the data set,
    - and prompt the user to specify area limits.
Pressing the ENTER key prior to changing the area limits permits
the selection of the entire data set.  Note that selections are
only available in the projection space of the data set.

To obtain additional documentation on the data set (i.e., data
contributor, processing parameters, etc.) press the help key --
"?" or any function key.

.
    .       (many help sections omitted)
.

*\grids\isores_eqv
begin constant
    file_title char 4-km%Isostatic%Residual%Grid%of%U.S.
    data_type char image
    number_of_rows short 772
    number_of_columns short 1300
    number_of_bands short 1
    missing_flag float 999999
    minimum_value float    -224.0
    maximum_value float 104.0
```

```
    image_format char bsq
    map_projection short 3
    upper_map_y float   5627
    left_map_x float    -2574
    grid_size(x) float  4
    grid_size(y) float  4
    lower_map_y float   2539
    right_map_x float   2626
      grid_origin char lowerleft
end constant
*\grids\isoreg_eqv
begin constant
    file_title char 8-km%Isostatic%Gravity%Grid%of%U.S.
    data_type char image
    number_of_rows short 386
    number_of_columns short 650
    number_of_bands short 1
    missing_flag float 999999
    minimum_value float -283
    maximum_value float 369
    image_format char bsq
    map_projection short 3
    upper_map_y float 5627
    left_map_x float  -2574
    grid_size(x) float 8
    grid_size(y) float 8
    lower_map_y float 2539
    right_map_x float 2626
      grid_origin char lowerleft
end constant
*\grids\isotop_eqv
begin constant
    file_title char 8-km%Isostatic%Topo%Grid%of%U.S.
    data_type char image
    number_of_rows short 476
    number_of_columns short 740
    number_of_bands short 1
    missing_flag float 999999
    minimum_value float -5729
    maximum_value float 3687
    image_format char bsq
    map_projection short 3
    upper_map_y float 5987
    left_map_x float -2934
    grid_size(x) float 8
    grid_size(y) float 8
    lower_map_y float 2179
    right_map_x float 2986
      grid_origin char lowerleft
end constant
*\grids\seggrav_eqv
begin constant
    file_title char 4-km%Gravity%Grid%of%U.S.
    data_type char image
    number_of_rows short 772
    number_of_columns short 1300
    number_of_bands short 1
    missing_flag float 9999
    minimum_value float -339
    maximum_value float 95
    image_format char bsq
```

```
    map_projection short 3
    upper_map_y float 5627
    left_map_x float -2574
    grid_size(x) float 4
    grid_size(y) float 4
    lower_map_y float 2539
    right_map_x float 2626
      grid_origin char lowerleft
end constant
*\grids\canada10_eqv
begin constant
    file_title char 10-min%geoid%grid%of%Canada
      data_type char image
    number_of_rows short 181
    number_of_columns short 601
    missing_flag float -9999
      var_1_min float 42
      var_1_max float 72
      var_2_min float -142
      var_2_max float -42
      var_3_min float -50.4
      var_3_max float 45.5
      var_4_min float -100
      var_4_max float 11
      var_5_min float -100
      var_5_max float 2
      var_6_min float -100
      var_6_max float 0
      var_7_min float 1.3
      var_7_max float 2.0
    minimum_value float 42
    maximum_value float 72
    map_projection char lat/lon
    upper_map_y float 72
    left_map_x float -142
    grid_size(x) float 0.166667
    grid_size(y) float 0.166667
    lower_map_y float 42
    right_map_x float -42
    image_format char bip
      grid_origin char lowerleft
end constant
*\grids\rapp1deg\rapp1deg_eqv
begin constant
    file_title char Global%1-deg%Grid
      data_type char image
    number_of_rows short 180
    number_of_columns short 360
    missing_flag float 9999
      var_1_min float -89
      var_1_max float 90
      var_2_min float 0
      var_2_max float 359
      var_3_min float 1.0
      var_3_max float 16.0
      var_4_min float -270.0
      var_4_max float 303
      var_5_min float 1
      var_5_max float 57
      var_6_min float -7527
      var_6_max float 5731
```

```
      var_7_min float 1
      var_7_max float 1038
    minimum_value float -89
    maximum_value float 90
    map_projection char lat/lon
    upper_map_y float 90
    left_map_x float 0
    grid_size(x) float 1
    grid_size(y) float 1
    lower_map_y float -90
    right_map_x float 360
    image_format char bip
end constant
*\grids\rapp30m\rapp30m_eqv
begin constant
    file_title char Global%30-min%Grid
    data_type char point
    number_of_rows short 360
    number_of_columns short 720
    missing_flag float -9999
    minimum_value float -94
    maximum_value float -60
    image_format char bsq
    map_projection char lat/lon
    upper_map_y float 90
    left_map_x float 0
    grid_size(x) float 0.5
    grid_size(y) float 0.5
    lower_map_y float -90
    right_map_x float 360
    image_format char bip
      grid_origin char lowerleft
end constant
*\grids\sograv\sograv_eqv
begin constant
    file_title char Southern%Occean%(60-72%S.%Lat)%Grid
    data_type char image
    number_of_rows short 241
    number_of_columns short 3601
    missing_flag float -999
      var_1_min float -72
      var_1_max float -60
      var_2_min float -180
      var_2_max float 180
      var_3_min float -113
      var_3_max float 83
    minimum_value float -72
    maximum_value float -60
    image_format char bip
    map_projection char lat/lon
    upper_map_y float -60
    left_map_x float 0
    grid_size(x) float 0.10
    grid_size(y) float  0.05
    lower_map_y float -72
    right_map_x float 360
      grid_origin char lowerleft
end constant
```

```
*\grids\dgrav_eqv
begin constant
    file_title char 2.5-min%Gravity%Anomalies%Grid%of%N.%America
    data_type char image
    number_of_rows short 1824
    number_of_columns short 4302
    number_of_bands short 1
    missing_flag float  -9999
    minimum_value float -360
    maximum_value float 200
    image_format char bsq
    map_projection char lat/lon
    upper_map_y float 81.16667
    left_map_x float  170.0
    grid_size(x) float 0.04166667
    grid_size(y) float 0.04166667
    lower_map_y float 5.16667
    right_map_x float 349.25
end constant
*\grids\isostat_eqv
begin constant
    file_title char 2.5-min%Isostatic%Gravity%(milligals)
    data_type char image
    number_of_rows short 637
    number_of_columns short 1537
    number_of_bands short 1
    missing_flag float -9999
    minimum_value float -220
    maximum_value float 110
    image_format char bsq
    map_projection char lat/lon
    upper_map_y float 49.54167
    left_map_x float 232.0
    grid_size(x) float 0.0416667
    grid_size(y) float 0.0416667
    lower_map_y float 23.0
    right_map_x float 296.04167
end constant
*\grids\dnaggrav_eqv
begin constant
    file_title char 6-km%Gravity%Grid%of%N.%America
    data_type char image
    number_of_rows short 1495
    number_of_columns short 1430
    number_of_bands short 1
    missing_flag float -999
    minimum_value float -367
    maximum_value float 341
    image_format char bsq
    map_projection char STM
    upper_map_y float 9280
    left_map_x float -4480
    grid_size(x) float 6
    grid_size(y) float 6
    lower_map_y float 700
    right_map_x float 4500
      grid_origin char lowerleft_y
end constant
```

```
*\sat\geo44\geo44asc_eqv
begin constant
    ff_input_format char &\sat\geo44\geo44.bfm
end constant
*\sat\geo44\geo44des_eqv
begin constant
    ff_input_format char &\sat\geo44\geo44.bfm
end constant
```

(End of the example file.)