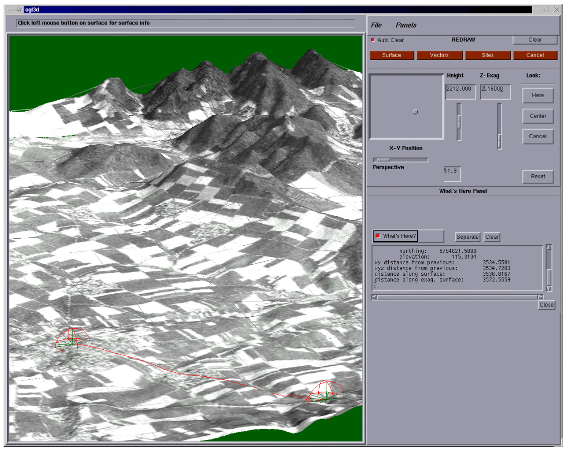
GRASS Reference Manual

Display Commands



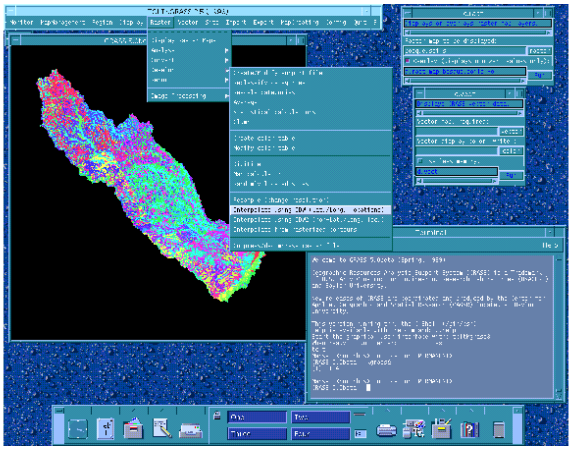
Click left mouse button on surface for surface info

Height: 2212.000

Z-Exag: 2.16000

Perspective: 11.9

```
         northing:    5764621.5000
         elevation:       115.3134
xy distance from previous:         3534.5581
xyz distance from previous:        3534.7283
distance along surface:            3536.8167
distance along exag. surface:      3572.5559
```



GRASS Development Team

USA Headquarters
Center for Applied Geographic & Spatial Research
Baylor University
P.O. Box 97351
Waco, Texas 76798-7351
USA

European Headquarters
Institute of Physical Geography-Landscape Ecology
University of Hannover
Schneiderberg 50
30167 Hannover
Germany

grass@baylor.edu

http://www.baylor.edu/~grass
http://www.geog.uni-hannover.de/grass/

Table of Contents

Topic Page

GRASS Introduction

GRASS (Geographic Resources Analysis Support System) is a raster based GIS, vector GIS, image processing system, and graphics production system. GRASS contains over 200 programs and tools to render maps and images on monitor and paper; manipulate raster, vector, and sites data; process multi-spectral image data; and create, manage, and store spatial data. GRASS uses both an intuitive windows interface as well as command line syntax for ease of operations. GRASS can interface with commercial printers, plotters, digitizers, and databases to develop new data as well as manage existing data.

GRASS is ideal for use in engineering and land planning applications. Like other GIS packages, GRASS can display and manipulate vector data for roads, streams, boundaries, and other features. GRASS can also be used to keep maps updated with its integral digitizing functions. Another feature of GRASS is its ability to use raster, or cell, data. This is particularly important in spatial analysis and design. GRASS functions can convert between vector data to raster data for seamless integration.

GRASS' strengths lie in several fields. The simple user interface makes it an ideal platform for those learning about GIS for the first time. GRASS is capable of reading and writing maps and data to many popular commercial GIS packages including ARC/Info and Idrisi. Users wishing to write their own code can do so by examining existing source code, interfacing with the documented GIS libraries, and using the GRASS Programmers Manual. This allows more sophisticated functionality to be integrated in GRASS.

The ability to work with raster data gives GRASS the unique ability to function as a surface modeling system. GRASS contains more than 100 multi-function raster analysis and manipulation commands. Surface processes such as rainfall-runoff modeling, flowline construction (as shown), slope stability analysis, and spatial data analysis are just a few of the many applications of GRASS to engineering and land planning. Since many of the raster tools are multi-functional, users can create their own maps from GRASS data analysis.

In addition to standard two-dimensional analysis, GRASS allows users to view data in three-dimensions. Raster maps, vector maps, and sites data can be used for visualization. Example applications of such capabilities include airspace analysis for airport planning (as shown), terrain analysis and “flybys”, and spatial trends. Tools in GRASS allow the user to animate any spatial data available with options to switch between data layers “on-the-fly”. Data used in 3-D visualization may also be saved as still pictures, or as mpeg movie files for later replay and analysis.

Accompanying its land planning and engineering applications, GRASS contains a suite of tools to aid in hydrologic modeling and analysis. Currently, tools are also available for performing such functions as watershed analysis, curve number generation, flood analysis, and stream channel characteristics for comprehensive watershed modeling. Other GRASS programs can generate graphs, statistics, and charts of modeled and calibrated data. Additionally, GRASS can use field data for model input or simulate parameters based on numerical data.

In addition to the traditional command line version of GRASS, a new user interface, based on Tcl/Tk has been written. This puts the power of spatial analysis and modeling into an easy to use Graphical User Interface that is platform-independent. This intuitive user interface lets users quickly and easily view, manipulate, and use data. Nearly all of the programs available in GRASS are available in the new GUI, with the standard command-line still available, giving users all of the functionality of GRASS.

This manual is part of a comprehensive set of documentation written to support GRASS. This Users Guide consists of a complete set of command references for all current GRASS functions and tools, including examples. An installation guide and fact sheet guides users through the installation process. For those wishing to write their own spatial analysis and modeling applications for GRASS, a Programmers Guide is also available. GRASS runs on a variety of UNIX and Linux platforms including SUN SPARCstations and Ultras, HP, Silicon Graphics, and PC’s running Windows 95 and Windows NT.

The GRASS Development Team is currently working to further upgrade and enhance the capabilities of GRASS. Future developments include tools that give the user the ability to work completely in 3-D, a capability that does not exist in any other GIS package. Users will be able to work with raster elevation data as well as vector and sites data in the 3-D environment, adding to the visualization capabilities of GRASS. Enhancements in the numerical processing functions of GRASS also now allow for floating-point operations to be performed on data.

For the latest information on GRASS contact the GRASS Development Team at grass@baylor.edu or visit our web sites at:

http://www.baylor.edu/~grass if you're in the U.S.

http://www.geog.uni-hannover.de/grass if you're in Europe

Look for our worldwide mirrors!

The GRASS Development Team is:

Bruce Byars and Markus Neteler are the development team leaders and coordinators.

Helena Mitasova and Bill Brown of the GMS Lab at UIUC have made significant contributions with the development of GRASS 5.

Additional authors include:
Lisa Zygo, Edward Zarecky, Jacques Bouchard, Steve Clamons, Brent Duncan, Jason Cipriano, Jim Westervelt, Michael Shapiro, Darrell McCauley, Dave Gerdes, Bill Hughes, Bernhard Reiter, Brook Milligan, Eliot Cline, Jaro Hofierka, Clay Cockrell, and Bob Lozar. See the web pages for author affiliations.

Note:

Many other people have contributed to the GRASS GIS. Without any one of them, GRASS would not exist in its current form. The authors of the individual programs are listed at the end of their manual page in the GRASS users manual, however, numerous authors of bug fixes and enhancements as well as people who have been working on coordination, integration, documentation and testing are not mentioned.

Please allow us to extend our most cordial thanks to all of you. If you contributed to GRASS at any point during its existence, let us know your name and e-mail address so we can add your name to the comprehensive on-line list.

To reference GRASS:

GRASS Development Team, 1999, Geographic Resources Analysis and Support System - GRASS: Baylor University, Waco, Texas.

GRASS Development Team
Center for Applied Geographic and Spatial Research
Baylor University
P.O. Box 97351
Waco, Texas, U.S.A. 76798-7351

d.3d

NAME
d.3d **-** Displays three-dimensional images based on raster map layers.
(GRASS Display Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
d.3d
d.3d help
d.3d [-l0an] map=name elevation=name [from_coordinate=x,y,z] [to_coordinate=x,y,z] [exaggeration=value] [lines=value] [field=value] [color=name] [box=name]

DESCRIPTION
d.3d displays three-dimensional graphic images based on GRASS raster map layers. The user identifies the viewing point, the line of sight, a vertical exaggeration factor, the viewing angle (field of view), the frequency and color of vector grid lines to appear in the display, the map to be displayed in 3-d, and the map whose category values are to be used as elevation values in the 3-dimensional image.

The program will be run non-interactively if the user specifies all needed parameter values and flag settings on the command line. Alternately, the user can simply type *d.3d* on the command line; in this case, the program will prompt the user for parameter values and flag settings using the standard interface described in the manual entry for *parser*.

Three-dimensional images can also be generated through the GRASS *d.display* program.

OPTIONS
Flags:
-*l* Display lines only, without displaying the raster base map.

-*0* Show zero elevations.

-*a* Each grid-cell is rendered using four elevation coordinates: one for each corner. By default, each corner takes the value of the grid-cell to the lower right. With the -*a* option each corner is calculated to be an average of the four bordering cell elevation values.

-*n* Show null elevations.

Parameters:
map=name The raster map layer used to generate the color shown in the 3-d output (i.e., the map whose x and y values will be displayed, using the z values present in the elevation map layer.)

elevation=name The raster map used to generate the texture in the 3-d image (i.e., the map whose category values will become the elevation values in the 3-d map displayed). This need not be an elevation map layer, although elevation is commonly used.
Default: elevation

from_coordinate=x,y,z Coordinates of the viewing point, given as: northing,easting,elevation. The default from_coordinate value is appropriate for the spearfish sample data set.
Default: Calculated at run-time just south-west of the south-west corner.

to_coordinate=x,y,z Coordinates of center of view, given as: northing,easting,elevation. The default to_coordinate value is appropriate for the spearfish sample data set.
Default: Calculated at run-time as the center of the image

exaggeration=value Vertical exaggeration factor. This value is multiplied by the "elevation" values (category values) in the elevation map layer. As the vertical coordinates are exaggerated, the elevation (z) value of the center of view (the to_coordinate) should be increased.
Default: 2.0

lines=value North-south and east-west trending lines can be drawn on the 3-d image output, to enhance the 3-d effect. The lines value specifies the number of rows and columns to skip between lines. 0 means "display no lines". Lines will be drawn in the color stated. The user can elect to only display these lines, without displaying the map, by setting the -*l* flag.
Default: 1

field=value The field of view (viewing angle) in the image, stated in degrees.
Default: 30

color=name Color of vector lines drawn on the output.
Options: color, white, red, orange, yellow, green, blue, indigo, violet, gray, black
Default: gray

The resulting three-dimensional image is drawn in the active frame on the graphics monitor. The user should select and erase the full graphics monitor frame before running *d.3d* to prepare the screen for graphics. Refer to the GRASS program *3d.view.sh* for a demonstration of the *d.3d* program.

INTERACTIVE MODE

The interactive mode prompts for the raster map to be rendered, the raster map to be queried for elevation information, and an optionally saved set of saved viewing parameters. The user is then presented with a form in which the options can be modified.

It is suggested that the following procedure be used to efficiently find the correct viewing parameters. Start with a gross grid (by default, a resolution is chosen which limits the total number of rows and columns), display lines only, set line colors to "color" (lines take their color from the map), set the box color to something other than "none", and use the default viewing coordinates. Experiment with the viewing coordinates until the map is positioned in a desirable configuration. When the map is lined up correctly, set the display resolution as low as you like for a final image.

BUGS

Several additions to this program have been suggested. These include the following:
1) Border the image with a visually appealing curtain.
2) Add x,y,z coordinates and scale information to the displayed 3-d image.
3) Provide a graphic-oriented user interface for identifying viewing parameters.
4) Display output more quickly by doing mathematics in integers.
5) Add the option of plotting vector map layers.

SEE ALSO

3d.view.sh, d.display, d.erase, d.frame, parser

AUTHOR

James Westervelt, U.S. Army Construction Engineering Research Laboratory

d.6386.delete

NAME

d.6386.delete **-** Deletes images from a PC6386 workstation hard disk, that were created by *d.6386.save*.
(SCS GRASS Graphics File Management Program)
(Available for PC6386 UNIX workstations, only)

GRASS VERSION

4.x

SYNOPSIS

d.6386.delete
d.6386.delete help

DESCRIPTION

This program allows a user to interactively remove PC6386 saved images from the hard disk. *d.6386.save* will store an image produced by GRASS display programs like *d.display*, *d.rast*, etc. from the graphics screen to the 6386 hard disk.

NOTES

This program is available for PC6386 UNIX workstations, only.

SEE ALSO

d.6386.save, *d.6386.show*

AUTHOR

P.W. Carlson, USDA, SCS, NHQ-CGIS

d.6386.save

NAME
d.6386.save **-** Stores screen images to the PC6386 hard disk.
(SCS GRASS Display Program)
(Available for PC6386 UNIX workstations, only)

GRASS VERSION
4.x

SYNOPSIS
d.6386.save
d.6386.save help

DESCRIPTION
This program allows a user to interactively save PC6386 screen images produced by GRASS display programs (*d.display*, *d.rast*, etc.) to the PC6386 hard disk. It will assign the extension ".GRS" to the filename given, and will store the image file in the user's current directory. The files are binary image files, representing the red, green, and blue planes of the display board graphic memory.

NOTES
This program is available for PC6386 UNIX workstations, only.

SEE ALSO
d.6386.delete, d.6386.show

AUTHOR
P.W. Carlson, USDA, SCS, NHQ-CGIS

d.6386.show

NAME

d.6386.show **-** Reads screen images stored on the PC6386 hard disk by *d.6386.save*.
(SCS GRASS Display Program)
(Available for PC6386 UNIX workstations, only)

GRASS VERSION

4.x

SYNOPSIS

d.6386.show
d.6386.show help

DESCRIPTION

Allows a user to interactively view PC6386 screen images produced by GRASS display programs (*d.display, d.rast*, etc.) and saved to the PC6386 hard disk by *d.6386.save*.

NOTES

This program is available for PC6386 UNIX workstations, only.

SEE ALSO

d.6386.delete, d.6386.save

AUTHOR

P.W. Carlson, USDA, SCS, NHQ-CGIS

d.ask

NAME
d.ask **-** Prompts the user to select a GRASS data base file from among files displayed in a menu on the graphics monitor.
(GRASS File Management Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
d.ask help
d.ask element=name,description [prompt="message"]

DESCRIPTION
d.ask is designed for shell scripts that need to prompt the user for the name of a data base file in the user's current GRASS mapset search path. After *d.ask* is invoked with needed parameters, the mouse becomes active and a menu containing files of the specified element type is displayed on the user's graphics monitor. The user is prompted by the prompt "message" to select one of the listed file names with the mouse. Specifically, the query that appears to the user takes the form:

```
Double click on the prompt "message"
Double click here to cancel
```

(A list of files of the specified element type from each of the mapsets listed in the user's mapset search path is also displayed.)

After the user responds, the mouse is deactivated and the displayed menu is erased from the screen (leaving any underlying materials on display intact). Three lines are written to standard output (the user's terminal screen):

```
name=file_name
mapset=mapset_name
fullname=file_name@mapset_name
```

Parameters:
element=name,description Name of a GRASS data base element, followed by a one word description of the element. GRASS data base elements are mapset subdirectories; these include: bdlg, cats, cell, cell_misc, cellhd, colr, colr2, dig, dig_ascii, dig_att, dig_cats, dig_plus, dlg, group, hist, icons, noise, paint, site_lists, and windows. The description will be used to display an error message to the screen if no files of the named element type exist in the user's mapset search path. The prompt "message" will appear in the pop-up menu displayed on the user's graphics monitor. The files listed in the menu will be of the specified element type, and exist in mapsets listed in the user's current mapset search path.

prompt="message" A brief message with which the user will be prompted. If this message contains more than one word, it should be enclosed within double quotes (""). It is a non-selectable message displayed in the menu bar of a pop-up menu. Selectable items in the displayed menu include a cancel option and any of the files in the mapset subdirectory name in the user's mapset search path.

EXAMPLE
Given the following input,

d.ask element=cell,raster prompt="raster map layer to be used" *d.ask* will prompt the user to select a raster (cell) file from among those listed in the "cell" directories of the mapsets listed in his current mapset search path. A pop-up menu will be displayed on the user's graphics monitor, containing the prompt:

```
        Double click on the raster map layer to be used
        Double click here to cancel
```

OUTPUT

If the user selects a file name from the displayed menu, *d.ask* writes three lines to standard output:

```
        name=file_name
        mapset=mapset_name
        fullname=file_name@mapset_name
```

The output is in the form of /bin/sh commands to set the variable name to the file name specified by the user (of the element type requested by *d.ask*), mapset to the GRASS mapset in which this file resides, and fullname to the name of the file and its mapset.

If the user elects not to select a file but instead chooses the "Double click here to cancel" option, the mouse will be deactivated and no variable assignments will be returned to standard output.

If no files of the specified element type are found in the user's current mapset search path, the mouse is activated and the following message is displayed on the user's graphics monitor:

```
        No element description files found
        Click here to continue
```

NOTES

Parameter values (the element name and description, and prompt message) cannot be supplied to *d.ask* interactively; they must be supplied on the command line or from a file.

SEE ALSO

g.ask, g.filename, g.findfile, g.gisenv, g.mapsets

BUGS

This program calls other GRASS programs which may generate confusing error messages. Example: an error message from the program *d.menu* will appear if a nonexistent map element is chosen.

AUTHOR

Michael Shapiro, U.S. Army Construction Engineering Research Laboratory

d.colormode

NAME
d.colormode **-** Allows the user to establish whether a map will be displayed using its own color table or the fixed color table of the graphics monitor.
(GRASS Display Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
d.colormode
d.colormode help
d.colormode mode=name

DESCRIPTION
d.colormode establishes what color table will be used to display maps to the graphics monitor.

Parameters:

mode=name	Options: fixed or float

Each time a new raster map layer is drawn on the screen, a color table associated with that map is loaded into the graphics display monitor. The command *d.colormode mode=fixed* requests that the monitor's color look-up table be fixed (i.e., static). The effect is that colors drawn on the screen by graphics calls will not change when subsequent maps are drawn to the screen. When the fixed option is used, the colors in the map's color table are mapped to the nearest colors available in the monitor's fixed color table. When the float option is used, the map's color table is loaded directly into the monitor's color look-up table.

There are advantages to using each color display mode. The *d.display* and *d.colors* programs allow the user to interactively modify the color tables of maps displayed to the graphics monitor. This is done by allowing the user to directly modify the monitor's color look-up table. Hence, to accommodate this option in *d.display* and *d.colors*, the *d.colormode mode=float* option is chosen. However, this option has the sometimes undesirable effect of changing the colors in which other maps are subsequently displayed to the graphics monitor (although it does not actually change the color tables of these latter maps). The float colormode is therefore best used when the user wishes to interactively change a map's color table, or when the user wishes to display one or more maps having the same color table. The fixed option allows any number of maps to be displayed to the screen, where each map uses different colors, but all use the same fixed color look-up table. You cannot toggle a map's color table when running in fixed mode.

This program will be run non-interactively if the user specifies the color mode on the command line (e.g., by typing *d.colormode mode=fixed* or *d.colormode mode=float*). Alternately, if the user simply types *d.colormode* on the command line, the program will prompt the user for the color mode using the standard GRASS *parser* interface described in the manual entry for *parser*.

NOTES
Some devices support only a fixed color look-up; e.g., ink-jet printers, plotters, and the AT&T 6300 (running a DEB board). The number of color categories capable of being displayed is also device-dependent; if your colors fall outside this range when in float mode, they may not be displayed in the colors indicated in the map's color table file. However, the color table files themselves will accurately represent the user's changes. Color table files associated with raster map layers are stored in the user's current mapset under the colr and colr2 directories.

BUGS

It is strongly recommended that the user erase the graphics monitor screen (e.g., by running *d.erase*) immediately after changing the mode between fixed and float.

SEE ALSO

d.colors, d.colortable, d.display, d.erase, d.rast, parser

AUTHOR

James Westervelt, U.S. Army Construction Engineering Research Laboratory

d.colors

NAME
d.colors **-** Allows the user to interactively change the color table of a raster map layer displayed on the graphics monitor.
(GRASS Display Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
d.colors
d.colors help
d.colors map=name

DESCRIPTION
A color table file associates specific colors with the categories of a raster map layer. The user can change these map category color assignments (i.e., change the map's color table) interactively, by first displaying the raster map to the graphics monitor and then running the program *d.colors*. If you have first set *d.colormode* to "float" before displaying the map to the graphics monitor, any color changes that you subsequently make while in *d.colors* will immediately (and interactively) appear on the graphics display.

While running *d.colors*, the cursor must be inside the window where the execution takes place. When in "float" colormode, the correct colors appear on the graphics monitor only if the cursor is inside the monitor. To see the correct colors, move the cursor from the execution window to inside the graphics monitor, then move back to the window to continue making changes.

If *d.colormode* is "fixed", any color changes made using *d.colors* will not immediately be shown on the graphics display; however, any color changes saved will still alter the map's color table and will appear next time the raster map layer is redisplayed (see *d.colormode* and *d.colortable*). Note that some graphics display monitors or drivers cannot support interactive color change.

The user must first display the relevant raster map layer to the active frame on the graphics monitor (e.g., using *d.rast* or *d.display*) before running *d.colors*. The user can then either enter the name of the raster map layer whose color table is to be changed on the command line (e.g., by typing: *d.colors map=soils*), or type *d.colors* without program arguments. If the user simply types *d.colors* without program arguments on the command line, *d.colors* will ask the user to enter the name of an existing raster map layer using the standard GRASS interface described in the manual entry for *parser*.

In either case, the user is then presented with the *d.colors* command menu, shown below. This menu is the same as the category and color changing portion of the *d.display* menu. The *d.colors* commands are listed beneath the Category Pointer Movement, Color Modification, Replotting Screen, and Quitting sections below. Commands are invoked by typing in the single-key response shown to the left below. (Longer descriptions of these commands appear to the right.) Results from invoking these commands will be reflected in the Category and Category Number sections of the *d.colors* screen. On the *d.colors* screen menu, commands appear in the right half of the screen, and the current status of categories appears in the left half of the screen.

CATEGORIES
0 No Data
1 (Category 1 description)
2 (Category 2 description)
. ....

```
.    ....
  CATEGORY NUMBER:
RED   0   0%
GREEN0   0%
BLUE  0   0%
  Shift Incr:   10   3%

  CATEGORY POINTER MOVEMENT
D/d  down (cats) Move pointer to next category
U/u  up   (cats) Move pointer to previous category

  COLOR MODIFICATIONS
R/r  RED    Increase/decrease RED intensity
G/g  GREEN  Increase/decrease GREEN intensity
B/b  BLUE   Increase/decrease BLUE intensity
I/i  increment   Increase/decrease increment (of intensity shift)
h    highlight   Highlight current color
+/-  shift colorsShift entire color table (up/down)
c    save color  Save color table
t    toggle tableToggle to different color table

  REPLOTTING SCREEN
*    Replot screen    Replots the screen

  QUITTING
Q    quit   Quits program
```

Changing categories **-** The keys d, D, u, and U are used to move to a different category. The lower case letters move up, u, and down, d, the category list one category at a time. The upper case letters move 10 categories at a time for fast movement. The cursor does wrap between the first and last categories. The current category is noted on the text screen with an arrow, and is indicated on the graphics screen by a box around the current color.

Changing colors **-** The color associated with the current category can be changed with the R, r, G, g, B, and b keys. The upper case letters increase the intensities of red R, green G, and blue B for the current category; the lower case letters decrease the intensities of these same colors for the current category. Video devices make all the colors of the spectrum by mixing red, green, and blue. For those accustomed to red, yellow, and blue being the primary colors, this can be confusing. For starters, yellow is made by mixing red and green. The intensities are listed on the text screen in as percentages.

Keys I and i increase and decrease the percentage change that each keystroke of one of the color keys (R, r, G, g, B, b) causes in its respective color. The increase increment is initially set to 10%. Thus, pressing the R key would will increase the red component of the current category by 10%.

Highlight **-** The h key toggles between the current category color and the current highlight color. This color is initially black but can be modified as above while in highlight mode. Blinking can be accomplished by repeatedly striking the h key. When changing to different categories using the movement keys as described above, while in highlight mode the category colors will be always left showing their actual colors. Only one category is highlighted at any one time.

Saving the current color table **-** Pressing the c key will save the current color table as you have modified it. This table will then be used next time you display or paint this raster map layer.

Color table toggle **-** Different types of color tables are suitable for different raster map layers. The key t flips between the following color tables: red, green, blue color ramp; gray scale; smooth changing color wave; random colors; and the saved color table.

Color table shift **-** The entire table is shifted up and down using the + and **-** keys.

Quitting the *d.colors* program **-** Pressing the Q key will cause you to quit the *d.colors* program. If colors have been modified but not saved, *d.colors* will ask:

```
Colors changed
Save the changes?  (y/n)
```

The user should type in y to save changes, or n to not save changes, before quitting the program. If the user types n, the program will ask:

```
Quit anyway?  (y/n)
```

NOTES

To see map color changes reflected on the display monitor as you make them, you must put the monitor in float color mode before running *d.colors*. It is also wise to erase the display screen after changing the color mode.

The map whose color table is to be altered with *d.colors* must already be on display in the active display frame on the graphics monitor before *d.colors* is run. This can be done using the command *d.rast map=name* (where name is a raster map layer whose color table the user wishes to alter).

The user might type the following sequence of commands to interactively change the colors of a raster soils map:

d.colormode mode=float
d.erase
d.rast map=soils
d.colors

After the user has saved any color changes made with *d.colors* and exited the program, the user could then redisplay the soils map with the new colors by typing:

d.colormode mode=fixed
d.erase
d.rast map=soils

Some color monitors may not support the full range of colors required to display all of the map's categories listed in the map's color table. However, regardless of whether the user can see the color changes he is effecting to a map's color table, any changes to a map's color table made with *d.colors* that are saved will appear in the map's color table.

Some monitors may not support an interactive color change capability.

SEE ALSO

d.colors.boxes, d.colormode, d.colortable, d., d.rast, p.colors, r.colors, parser

AUTHOR

James Westervelt, U.S. Army Construction Engineering Research Laboratory

d.colortable

NAME
d.colortable **-** To display the color table associated with a raster map layer.
(GRASS Display Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
d.colortable
d.colortable help
d.colortable map=name [color=name] [lines=value] [cols=value]

DESCRIPTION
The GRASS program *d.colortable* is used to display the color table associated with a raster map layer in the active frame on the graphics monitor. The map name should be an available raster map layer in the user's current mapset search path and location.

Parameters:
map=name Name of a raster map layer in the user's current mapset search path whose color table is to be displayed.

color=name Color of the lines separating the colors of the color table.
Options: red, orange, yellow, green, blue, indigo, violet, gray, white, and black.
Default: white

lines=value Number of lines to appear in the color table.
Options: 1 **-** 1000

cols=value Number of columns to appear in the color table.
Options: 1 **-** 1000

If the values of both lines and cols are not specified by the user, *d.colortable* divides the active frame equally among the number of categories present in the named raster map layer. If one option is specified, the other is automatically set to accommodate all categories. If both are specified, as many categories as possible are displayed.

The user can specify all needed parameters on the command line using the form:

```
d.colortable map=name [color=name] [lines=value] [cols=value]
```

If the user specifies the name of a map on the command line but does not specify the values of other parameters, parameter default values will be used. Alternately, if the user types simply *d.colortable* on the command line without any program arguments, the program will prompt the user for parameter settings using the standard GRASS *parser* interface described in the manual entry for *parser*.

EXAMPLE
The user running the command:

```
d.colortable map=soils color=red lines=1 cols=3
```

would see the active graphics frame divided into three columns extending the full frame height. The lines dividing the color table associated with the soils map would be displayed in red. The user would see, at most, only three of the colors from the soils color table displayed in the active frame (because the user requested that this frame be divided into three sections).

NOTES

If the user wishes to display the entire color table associated with a map, the user should either stipulate a number of lines (rows) and columns (cols) sufficient to accommodate the number of categories in the map's color table, or fail to assign values to one or both of lines and/or cols. If the user runs *d.colortable* using the default number of lines and columns (the full graphics frame), all categories from the map's color table will be displayed. However, if the user requests that the color table associated with a map which has 10 data categories be displayed in a graphics frame with only 3 lines (rows) and 2 columns (a total of six cells), only six of the ten map categories will be displayed.

The user should run the command *d.colormode mode=float* before running *d.colortable*, for accurate display of the map's color table.

The user should run the GRASS program *d.erase* between runs of *d.colortable* to avoid confusion.

SEE ALSO

d.colormode, d.colors, d.display, d.erase, d.rast, parser

AUTHOR

James Westervelt, U.S. Army Construction Engineering Research Laboratory

d.display

NAME
d.display **-** A menu-driven, highly interactive display program for viewing maps and producing final map products
(GRASS Display Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
d.display

DESCRIPTION
The GRASS program *d.display* is used to display maps on a graphics monitor and prepare final map products for printing. It is advisable to first run *d.erase* before each run of *d.display* to prepare the screen for graphics.

After invoking the command *d.display*, the DISPLAY MAIN MENU appears on the monitor. Options in this and subsequent *d.display* menus are selected through use of the pointing device (mouse). When the mouse cursor is over the desired menu selection, the user simply presses any button on the mouse.

d.display is a macro that allows the user to access a wide range of GRASS display functions in a menu-driven and highly interactive environment. Most of the functions accessible through *d.display* can also be run independently of one another by running the GRASS display (*d.*) commands listed in the **SEE ALSO** section, below. However, in many cases, these other *d.* commands give the user greater flexibility and make available more options than are available through the *d.display* menus. For example, the *d.display* label file option runs the *d.label* command using defaults. Users running *d.label* directly from the command line have the choice of stipulating specific label background and text colors, as well as text sizes and font types. Users running the label file option through *d.display* cannot specify these options; instead, default values are used (see *d.label* for default values used).

The *d.display* tool is best learned by playing with its options. Refer to the individual *d.* and other commands in the **SEE ALSO** section for a detailed examination of these functions.

NOTES
The user can also modify the current region settings from the DISPLAY MAIN MENU. Using the mouse, the user can choose to zoom in on a portion of the displayed raster map layer to create a smaller geographic region. Once the user has defined the corners of this area using the mouse, the smaller area is enlarged and replotted on the screen. The user can similarly choose to zoom out. In this case, the user uses the mouse to encompass that portion of the reduced map that the user wishes to make the current geographic region and display in the center of the screen. The larger map is then replotted and displayed. Finally, the user can enter geographic coordinates for the region's corners from the keyboard using the GRASS command *g.region*. This is done by selecting the type coordinates option.

The main menu also offers the user the option of hiding the main menu for ten seconds (hide menu for 10 seconds).

The *d.display* command must be run in conjunction with a program which converts the GRASS graphics commands generated by *d.display* to the graphics commands that the current device understands. This device will almost always be a graphics monitor. On most systems the user is required to start up this other program separately from (and prior to) *d.display*. In such cases, a locally-defined program, usually "monitor", is used (see your GRASS manager). (For example, on MASSCOMP machines, the *d.mon* program must be running in the foreground on the graphics monitor. On SUN machines, the *d.mon* program can be started from any terminal and run in the background.)

d.display sends program output to the current graphics frame (set, by default, to the full graphics screen unless modified by the user). You can use *d.frame* to create and choose a smaller display frame on the graphics monitor for use by *d.display*.

See the GRASS macro *3d.view.sh* stored under $GISBASE/scripts for an example of how you can create a *d.display*-like macro using a shell script.

SEE ALSO
d.3d, d.colormode, d.colors, d.frame, d.label, d.legend, d.measure, d.mon, d.rast, d.scale, d.sites, d.vect, d.zoom, g.region, p.menu, v.digit
See *3d.view.sh* or *slide.show.sh* under $GISBASE/scripts for examples of *d.display*-like macros.

AUTHOR
James Westervelt, U.S. Army Construction Engineering Research Laboratory

d.erase

NAME

d.erase **-** Erases the contents of the active display frame on the user's graphics monitor.
(GRASS Display Program)

GRASS VERSION

4.x, 5.x

SYNOPSIS

d.erase
d.erase help
d.erase [color=name]

DESCRIPTION

d.erase erases the contents of the active graphics frame, and replaces it with the color black (by default) or by whatever color is specified by the user. *d.erase* will not alter the assignment of the active frame.

Parameter:
color=name Color with which active frame will be erased.
Options: red, orange, yellow, green, blue, yellow, indigo, violet, black, white, gray, brown, and magenta
Default: black

SEE ALSO

d.frame, d.mon

AUTHOR

James Westervelt, U.S. Army Construction Engineering Research Laboratory

d.font

NAME
d.font **-** Selects the font in which text will be displayed on the user's graphics monitor.
(GRASS Display Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
d.font
d.font help
d.font font=name

DESCRIPTION
d.font allows the user to select use of a specific text font for display of text on the graphics monitor. The GRASS program show.fonts.sh is a UNIX Bourne shell macro which names and displays the fonts that can be selected using *d.font*. If the user does not specify a font when using other GRASS programs that display text, the font type romans is used by default.

The user can run this program either non-interactively or interactively. If the user specifies a font type name on the command line the program will run non-interactively. Alternately, the user can simply type *d.font* on the command line; in this case, the program will prompt the user for a display text font type using the standard GRASS interface described in the manual entry for *parser*.

Parameter:
font=name Name of a font type, from among the font types italicized below.
Default: romans
Options: (italized)
cyrilc - Cyrillic
gothgbt - Gothic Great Britain triplex
gothgrt - Gothic German triplex
gothitt - Gothic Italian triplex
greekc - Greek complex
greekcs - Greek complex script
greekp - Greek plain
greeks - Greek simplex
Italicc - Italian complex
italiccs - Italian complex small
italict - Italian triplex
romanc - Roman complex
romancs - Roman complex small
romand - Roman duplex
romanp - Roman plain
romans - Roman simplex
romant - Roman triplex
scriptc - Script complex
scripts - Script simplex

NOTES
The font type romans is the fastest font type to display to the graphics monitor.

SEE ALSO
d.INTRO, d.text, d.title, show.fonts.sh, parser

AUTHOR
James Westervelt, U.S. Army Construction Engineering Research Laboratory

d.font uses the public domain version of the Hershey Fonts created by Dr. A.V. Hershey while working at the U.S.National Bureau of Standards.

d.frame

NAME
d.frame **-** Manages display frames on the user's graphics monitor.
(GRASS Display Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
d.frame
d.frame help
d.frame [cepsD] [frame=name] [at=bottom,top,left,right]

DESCRIPTION
This program manages display frames on the user's graphics monitor. GRASS display programs at run-time connect with graphics rendering programs. While the display programs are identical on every hardware platform, the graphics rendering programs are (essentially the only GRASS programs) designed for individual hardware devices. These rendering programs are managed with the GRASS program *d.mon*. Graphics are displayed in rectangular frames on whatever graphics monitor the user is currently directing GRASS display output to. These frames are created and managed with this program, Note that GRASS frame contents are not retained when one frame covers another. You cannot shuffle frames from top to bottom and then back again. They simply define rectangular areas on the screen where subsequent drawing will occur. *d.frame*.

Flags:
-*c* Creates a new display frame on the graphics monitor.

-*e* Removes all existing display frames and reinitializes the entire graphics screen (the full-screen display frame).

-*p* Prints the name of the active frame, in which GRASS display output will appear.

-*s* Selects a frame for the display of GRASS graphics. This frame is then known as the "active frame".

-*D* Prints the status of the user's graphics monitor and active display frame to standard output. Information includes the name and the dimensions of the current frame on the graphics monitor, given in the form bottom top left right. This function is useful for debugging output, and for determining display screen coordinates.

Parameters:
frame=name The name of the display frame to be created/selected.

at=bottom,top,left,right Where to place the frame (implies -c). Frame coordinates are stated in the form: bottom,top,left,right. The lower-left corner of the graphics monitor always is at location 0,0 while the monitor's upper-right corner is always at 100,100.

NOTES
If the user has created multiple display frames that overlap one another, whatever the user displays in the active frame will overwrite those portions of the underlying frame where these frames overlap.

SEE ALSO
d.erase, d.mon

AUTHOR
James Westervelt, U.S. Army Construction Engineering Research Laboratory
Michael Shapiro, U.S. Army Construction Engineering Research Laboratory

d.geodesic

NAME
d.geodesic **-** Displays a geodesic line, tracing the shortest distance between two geographic points along a great circle, in a longitude/latitude data set.
(GRASS Display Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
d.geodesic
d.geodesic help
d.geodesic [coor=lon1,lat1,lon2,lat2] [lcolor=name] [tcolor=name]

DESCRIPTION
d.geodesic displays a geodesic line in the active frame on the user's graphics monitor. This line traces the shortest distance between two user-specified points on the curved surface of a longitude/latitude data set. The two coordinate locations named must fall within the boundaries of the user's current geographic region.

OPTIONS
This program can be run either interactively or non-interactively. If the user types *d.geodesic* on the command line without other program parameters, the mouse will be activated; the user is asked to use the mouse to indicate the starting and ending points of each geodesic line to be drawn. The default line color (white) and text color (red) will be used.

Alternately, the user can specify the starting and ending coordinates of the geodesic, line color, and text color on the command line, and run the program non-interactively.

Once the user indicates the starting and ending coordinates of the geodesic, the line and its length (in miles) are displayed to the user's graphics monitor.

Parameters:
coor=lon1,lat1,lon2,lat2 Starting and ending coordinates, in longitude and latitude values, of the geodesic line to be drawn.

lcolor=name Line color in which geodesic will be displayed.
Options: red, orange, yellow, green, blue, magenta, indigo, violet, gray, white, black
Default: white

tcolor=name Text color in which the length of the geodesic will be displayed.
Options: red, orange, yellow, green, blue, magenta, indigo, violet, gray, white, black
Default: red

NOTES
This program works only on GRASS data bases using a longitude/latitude coordinate system.

SEE ALSO
d.rhumbline

AUTHOR
Michael Shapiro, U.S. Army Construction Engineering Research Laboratory

d.graph

NAME
d.graph **-** Program for generating and displaying simple graphics to the graphics display monitor.
(GRASS Display Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
d.graph
d.graph help
d.graph [input=name] [color=name]

DESCRIPTION
d.graph draws graphics that are described either from standard input (default), or within a file (if an input file name is identified on the command line). If graphics commands are entered from standard input, a ctrl-d is used to signal the end of input to *d.graph*.

The program can be run interactively or non-interactively. The user can run the program completely non-interactively by specifying the name of a graphics file containing *d.graph* graphics commands and the values of all needed parameters on the command line. The user can instead elect to run the program partially interactively, by specifying any/all of the parameters except the graphics *file=name* parameter on the command line. In this case, *d.graph* will expect the user to input *d.graph* graphics commands from standard input (i.e., the keyboard) and will (silently) prompt the user for these graphics commands.

Alternately, the user can simply type *d.graph* on the command line, and be prompted for the values of all parameters (the user can still input graphics commands from either an input file or standard input using this form). In this case, the user is presented with the standard GRASS *parser* interface described in the manual entry for *parser*.

Parameters:
file=name Name of file containing graphics commands. This is a UNIX file name; the file must be located in the user's current working directory or be specified by its full path name. If no filename is specified commands are taken from the standard input.

color=name Starting color desired for graphics.
Options: red, orange, yellow, green, blue, indigo, violet, magenta, brown, gray, white, and black
Default: white

The coordinate system used is 0-100 in x and 0-100 in y, regardless of the graphics monitor display frame size and aspect. The (0,0) location is the lower left corner of the active graphics monitor display frame. All values may be floating point.

The graphics language is simple, and uses the following commands:

# comment A line of comment which is ignored in the processing.

move xpos ypos The current location is updated to xpos ypos. Values are stated as a percent of the active display frame's horizontal (xpos) and vertical (ypos) size, and may be floating point values. Values are between 0-100. Note. A space must separate xpos and ypos.

draw xpos ypos A line is drawn in the current color from the current location to the new location xpos ypos, which then becomes the current location. Values are stated as a percent of the active display frame's horizontal (xpos) and vertical (ypos) size, and may be floating point values. Values are between 0-100. Note. A space must separate xpos and ypos.

color color Sets the current color to that stated; subsequent graphics will be drawn in the stated color, until the current color is set to a different color.
Options: red, orange, yellow, green, blue, indigo, violet, brown, magenta, gray, white, and black.

size xper yper Subsequent text will be drawn such that the text is xper percent of the graphics monitor display frame wide and yper percent of the frame high. By default, the text size is set to 1 percent of the active frame's width and 1 percent of the frame's height if unspecified by the user (this may be too small to be seen by the user). Note. A space must separate xper and yper.

text line-of-text The stated text is drawn at the current location using the current color, and the new current location is then positioned at the end of the text string.

icon type size x y Draws an icon of types o, x, or + with specified size at location x,y.
Note: type o designates a square.

polygon

```
 xpos ypos
 xpos ypos
   .
   .
   .
```

The coordinates appearing beneath the word polygon, one pair per line, circumscribe a polygon that is to be filled with the current color.

EXAMPLE

For an example use of *d.graph*, examine the contents of the macro command $GISBASE/bin/grass.logo.sh located in the GRASS command bin (which draws a GRASS logo by inputting *d.graph* graphics commands that are stored in a shell file). Note that the coordinates in the *grass.logo.sh* macro were taken directly off an image drawn by hand on graph paper.

NOTES

d.graph remembers the last screen location (xpos ypos) to which the user moved, even after the user erases the display frame. If the user runs *d.graph* repeatedly, and wishes to start anew with the default (xpos ypos) screen location, the user should clear the display frame between runs of *d.graph*.

LIMITATIONS

There are no automated ways of generating graphic images. It is anticipated that GRASS user sites will write programs to convert output from a resident graphics editor into GRASS *d.graph* format.

SEE ALSO

d.INTRO, d.font, d.mapgraph, d.text, grass.logo.sh, parser

AUTHOR

James Westervelt, U.S. Army Construction Engineering Research Laboratory

d.grid

NAME
d.grid **-** Overlays a user-specified grid in the active display frame on the graphics monitor.
(GRASS Display Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
d.grid
d.grid help
d.grid size=value [color=name] [origin=easting,northing]

DESCRIPTION
d.grid overlays a grid of user-defined size and color in the active display frame on the graphics monitor. The grid will overlay, not overwrite, the contents of the active display frame.

d.grid can be run non-interactively or interactively. If the user specifies the grid size and (optionally) the grid color on the command line the program will run non-interactively; if no grid color is given the default will be used. Alternately, the user may simply type *d.grid* on the command line; in this case, the program will prompt the user for parameter values using the standard GRASS user interface described in the manual entry for *parser*.

OPTIONS
Parameters:
size=value Size of grid to be drawn in the active display frame, in current map coordinate system units. It should be noted that, depending on the scale of the map layer displayed in this frame, the grid may fall outside the user's viewing range.
Options: 0-100000

color=name Sets the current grid color to name.

origin Lines of the grid pass through this coordinate. The coordinate need not be within the current frame
Options: red, orange, yellow, green, blue, indigo, violet, magenta, brown, gray, white, and black.
Default: gray

NOTES
d.grid will not erase grids already displayed in the active graphics display frame by previous invocations of *d.grid*; multiple invocations of *d.grid* will therefore result in the drawing of multiple grids inside the active graphics frame. (A command like *d.erase*, which erases the entire contents of the active display frame, must be run to erase previously drawn grids from the display frame.)

Currently, the grid size can only be drawn in the units of the map coordinate system used by the current GRASS location.

SEE ALSO
d.display, d.erase, d.frame, d.legend, d.rast, d.scale, parser

AUTHOR
James Westervelt, U.S. Army Construction Engineering Research Laboratory

d.his

NAME
d.his **-** Produces and displays a raster map layer combining hue, intensity, and saturation (his) values from user-specified input raster map layers.
(GRASS Display Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
d.his
d.his help
d.his h_map=name [i_map=name] [s_map=name] [out=name]

DESCRIPTION
his stands for hue, intensity, and saturation. This program produces a raster map layer providing a visually pleasing combination of hue, intensity, and saturation values from two or three user-specified raster map layers.

The human brain automatically interprets the vast amount of visual information available according to basic rules. Color, or hue, is used to categorize objects. Shading, or intensity, is interpreted as three-dimensional texturing. Finally, the degree of haziness, or saturation, is associated with distance or depth. This program allows data from up to three raster map layers to be combined into one new raster map layer which retains the original information in terms of hue, intensity, and saturation.

OPTIONS
This program can be run non-interactively or interactively. It will run non-interactively if the user specifies on the command line the name of a map containing hue values (h_map), and the name(s) of map(s) containing intensity values (i_map) and/or saturation values (s_map). If the user also names an output map (out) on the command line, the combined hue, saturation, and intensity values will be saved as a raster map layer in the user's current mapset; otherwise, the resulting image will only be displayed in the active display frame on the graphics monitor.

Alternately, the user can run the program interactively by typing *d.his* without naming parameter values on the command line. In this case, the program will prompt the user for parameter values using the standard GRASS *parser* interface described in the manual entry for *parser*.

Parameters:
h_map=name Name of input raster map layer to be used for HUE values.

i_map=name Name of input raster map layer to be used for INTENSITY values.

s_map=name Name of input raster map layer to be used for SATURATION values.

out=name Name of the output raster map layer combining hue, intensity, and saturation values from input layers.

While any raster map layer can be used to represent the hue information, map layers with a few very distinct colors work best. Only raster map layers representing continuously varying data like elevation, aspect, weights, intensities, or amounts can suitably be used to provide intensity and saturation information.

For example, a visually pleasing raster map layer can be made by using a watershed map for the hue factor, an aspect map for the intensity factor, and an elevation map for saturation. (The user may wish to leave out the elevation information for a first try.) Ideally, the resulting image should resemble the view from an aircraft looking at a terrain on a sunny day with a bit of haze in the valleys.

THE PROCESS

Each map cell is processed individually. First, the working color is set to the color of the corresponding cell in the map layer chosen to represent HUE. Second, this color is multiplied by the red intensity of that cell in the INTENSITY map layer. This map layer should have an appropriate gray-scale color table associated with it. You can ensure this by using the color manipulation capabilities of *d.display* or *d.colors*. Finally, the color is made somewhat gray based on the red intensity of that cell in the SATURATION map layer. Again, this map layer should have a gray-scale color table associated with it.

NOTES

This program produces an image and (optionally) a raster map layer with 1000 colors (10 intensities each of red, green, and blue). The resulting image and raster map layers will not display properly if the graphics display monitor does not have at least 1000 colors while the user is running in float colormode. Thus, unless the display device has 1000 colors, it is necessary to run the GRASS command *d.colormode mode=fixed* before running *d.his*. Otherwise, the colors will be incorrectly displayed on the graphics monitor.

Either (but not both) of the intensity or the saturation map layers may be omitted. This means that it is possible to produce output images that represent combinations of his, hi, or hs.

SEE ALSO

d.3d, d.colormode, d.colors, d.colortable, d.display, hsv.rgb., rgb.hsv.sh, parser

AUTHOR

James Westervelt, U.S. Army Construction Engineering Research Laboratory

d.histogram

NAME
d.histogram **-** Displays a histogram in the form of a pie or bar chart for a user-specified raster file.
(GRASS Display Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
d.histogram
d.histogram help
d.histogram [-zqnC] map=name [color=name] [style=name][nsteps=value]

DESCRIPTION
d.histogram displays the category-value distribution for a user-specified raster map layer, in the form of a bar chart or a pie chart. The display will be displayed in the active display frame on the graphics monitor, using the colors in the raster map layer's color table. The program determines the raster file's category value distribution by counting cells.

OPTIONS
The user can run this program either non-interactively or interactively. The program will be run non-interactively if the user specifies the name of a raster map layer and (optionally) any other desired parameters and flags on the command line, using the form:

d.histogram [-zqnC] map=name [color=name] [style=name][nsteps=value]

If a map name is given on the command line, any other parameter values left unspecified on the command line will be set to their default values (see below). Alternately, the user can simply type *d.histogram* on the command line, without program arguments. In this case, the user will be prompted for needed inputs and option choices using the standard GRASS user interface described in the manual entry for *parser*.

Flags:
-*z* Display zero-data information (cells with category value zero) in the histogram. If the -*z* flag is set, then cells with category value 0 in the named raster map layer will be included in the pie or bar chart. If this flag is not set, then cells with category value 0 will be excluded from the bar chart.

-*q* Gather the histogram quietly, without printing messages on program progress to the user's terminal.

-*n* Display information for null cells.

-*C* Report for ranges defined in cats file (fp maps only).

Parameters:
map=name The name of an existing raster map layer in the user's current mapset search path for which a histogram is to be displayed.

color=name The name of the color to be used for the axis, text-labels, and tic-marks on the pie or bar chart.
Options: white, red, orange, yellow, green, blue, indigo, magenta, violet, brown, gray, and black
Default: color=white

style=name The style of the histogram. If style=bar, then *d.histogram* will displays the statistics in the form of a bar chart. If style=pie, then *d.histogram* will display the statistics in the form of a pie chart.
Options: bar or pie
Default: style=bar

nsteps=value Number of steps to divide the data range into (fp maps only).
Default: 255

NOTES

d.histogram uses the current geographic region settings and the current mask (if one exists).

d.histogram uses the colors in the map's color look-up table (i.e., the map's colr or colr2 file). To view results correctly on the display monitor, make sure the color mode is correctly set before running *d.histogram* (i.e., set *d.colormode=float*).

d.histogram does not erase the active frame before displaying output.

SEE ALSO

d.colormode, d.colors, d.colortable, d.display, d.erase, g.region, r.mask, r.stats, parser

AUTHOR

Dave Johnson, DBA Systems, Inc.

d.icons

NAME
d.icons **-** Displays points, as icons, at user-defined locations in the active display frame on the graphics monitor.
(GRASS Display Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
d.icons
d.icons help
d.icons [-r] icon=name [color=name] [size=value] [points=name]

DESCRIPTION
d.icons graphically displays point (site) locations as icons in the active frame on the graphics monitor. Geographic coordinates are read either from standard input or from an input file whose name is stated by the user. At each site location, a user-defined icon is displayed.

OPTIONS
The user must enter at least the name of an icon file storing a graphic representation of the icon to be displayed, and the geographic coordinates of the points at which they will appear.

The user can run this program either non-interactively or interactively. The program will be run non-interactively if the user specifies the name of an icon file and (optionally) any other desired parameters on the command line, using the form:

d.icons [-r] icon=name [color=name] [size=value] [points=name]

If the user fails to specify the name of a points file on the command line, the program will prompt the user to enter geographic coordinates from standard input. If the user specifies at least the name of an icon file on the command line, any other parameter values left unspecified by the user will be set to their default values (see below).

Alternately, the user can simply type *d.icons* on the command line, without program arguments. In this case, the user will be prompted for needed parameter values using the standard GRASS user interface described in the manual entry for *parser*. The geographic coordinates at which icons are to be displayed can still be input from an input file (points) in interactive use.

Flag:
-r Coordinates are input in reverse order (i.e., as northing easting).

Parameters:
icon=name The name of an existing file containing a graphic representation of the icon to be drawn. Icon files can be created by the user using the *p.icons* command and are stored under the icons directory under the user's current mapset.

color=name Sets the current icon color to the name stated.
Options: red, orange, yellow, green, blue, indigo, violet, gray, white, and black
Default: color=white

size=value   The icon scaling factor.
   Options: 1-1000
   Default: size=1

points=name   The name of a UNIX file containing the geographic coordinates of sites at which icons are to be drawn. Since this is a UNIX file, the user should specify the name using standard UNIX file naming conventions. If this file is not in the user's current working directory, its pathname should be specified. The file contents should consist of a series of geographic coordinates that fall within the boundaries of the current geographic region. Each site location should be stated on a separate line as an easting and northing (in that order) separated by a single blank space. If no points file is specified by the user, input is taken from standard input and should be given in the same form; to end standard input, type end (or press control-d).

EXAMPLE

An icons file contains a graphic representation of the icon to be displayed. Here, spaces represent areas of no color, and x's represent areas of color. For example, the user might construct an icon resembling a cross in the following way:

```
  x
  x
xxxxx
  x
  x
  x
```

This icon might be stored in a file called cross (under the user's $LOCATION/icons directory). If the user specified that this be the icon file used while running *d.icons*, this cross would then appear in the color and size specified by the user, at each site location named in the points file.
Note: icons are created with the *p.icons* command.

The points file lists the geographic coordinates of site locations (at which icons will be displayed). This file should take the form:

```
easting northing
easting northing
 ...
```

If the user sets the *-r* flag, the order of these coordinates should be reversed (i.e., coordinates should be given as northing easting).

If the points file is not specified, then the coordinates are read from the keyboard or across a pipe. This feature allows users to enter the coordinates by hand, or, more usefully, to get then from another program. For example, to display icons at locations specified in a sites list:

s.out.ascii name | d.icons icon=cross

where name is the name of a site list, and cross is the name of an icon.

SEE ALSO

d.points, d.sites, d.where, p.icons, s.db.rim, s.out.ascii, v.db.rim, parser

AUTHOR

Dave Johnson, DBA Systems, Inc.

d.igraph

NAME
d.igraph **-** Graphics interface for GRASS on the Intergraph.
(Map Development Tool)

GRASS VERSION
4.x

SYNOPSIS
d.igraph

DESCRIPTION
d.igraph is the GRASS interface to the GRASS graphics driver on the Intergraph workstations. GRASS can be run from an ascii terminal or from a window on the Intergraph workstation. There are some pitfalls to running GRASS from a window; these are discussed in the **NOTES** section below.

When *d.igraph* is run a pull-down menu will appear on the workstation screen and some information will be displayed on the ascii terminal. The pull-down menu, which works the same way other Intergraph pull down-menus work, will look like this:

```
 _______________________
|  GRASS Commands        |
|  start/select          |
|  stop                  |
|  GRASS Graphics        |
|  resize/move           |
|  virtual screens      >|
|  colors               >|
| EXIT                   |
|________________________|
```

The menu is divided into two categories: GRASS Commands and GRASS Graphics. GRASS Commands in the pull-down menu have GRASS command-line equivalents. These command-line equivalents can be used without running *d.igraph*, although users may find the pull-down menu commands easier and faster to use. On the other hand, there are no command-line equivalents for the GRASS Graphics commands in the pull-down menu. These commands, which must be run from inside *d.igraph*, enable you to modify the size, placement, and number of colors, of the graphics window.

The first command under GRASS Commands is start/select; selecting this command with the mouse will start the graphics driver IGRAPH and then select it for output. This pull-down menu command is equivalent to typing the following commands at the GRASS command line:

d.mon start=IGRAPH (start and select the IGRAPH graphics driver)

or

d.mon -*s start=IGRAPH* (start but don't select the IGRAPH driver)
d.mon select=IGRAPH (select the already started IGRAPH driver)

Using the start/select option saves the user from having to type the above command(s). After the graphics driver is started, the pull-down menu will disappear and the *d.igraph* program will exit, leaving the user at the GRASS prompt. When the second command (stop) under GRASS Commands is selected, the graphics driver IGRAPH will stop and the GRASS window will disappear. The equivalent GRASS command-line command is: *d.mon stop=IGRAPH.*

The second category (GRASS Graphics) in the pull-down menu contains three commands written for the GRASS graphics driver IGRAPH. These commands establish parameters for size, location, and number of colors the graphics driver will use to create a GRASS graphics window. Therefore, they should be run before starting the graphics driver with the start/select command. These parameters will not affect a GRASS window that has been started, but they will be saved and used the next time the graphics driver is started.

The first command in the second category is resize/move. When the resize/move command is selected, the pull-down menu will disappear and a special Resize window, colored grey, will be created on the graphics monitor. The Resize window will be the exact size and at the same location as the GRASS graphics window would be if the graphics driver had been started. Use the resize and move icons on the Resize window to modify the window. When the desired size and position has been achieved, delete the Resize window. To do this, move the mouse to the left side of the icon strip at the top of the Resize window and click the mouse on the small box containing an X. Note that the window size is displayed on the ascii terminal in which the *d.igraph* program was started. Now, each time the graphics monitor is started, the window will be the same size and at the same location as the Resize window.

The second command is virtual screens. This command has a sub-menu with a listing of virtual screen numbers (from 0 to 3). Of the virtual screen numbers listed (0 to 3), only one or two of those numbers will be active and selectable from the sub-menu. The number of selectable virtual screen numbers will depend on the type of workstation and its virtual screen configuration. When the IGRAPH driver is started, it will create a GRASS window on the virtual screen whose number was picked by this command. Note that the current virtual screen number is displayed on the ascii terminal n which the *d.igraph* program was started.

The third graphics command is colors. This command has a sub-menu with a listing of the possible range of active colors the workstations will support (16 to 512). The number of active colors is determined by the number of planes the graphics processor supports (9 planes will give 512 active colors). We do not suggest running GRASS with fewer than 256 active colors.

Each workstation can be configured differently with regard to the number of virtual screens and the available number of colors. A certain amount of testing will have to been done to determine what works best with a particular workstation.

The last command is the EXIT command. This command is used to leave the *d.igraph* program.

USING THE GRAPHICS WINDOW

Once the GRASS graphics driver is started, a window will appear on the screen. The GRASS graphics window is mouse- activated. The mouse must be within the GRASS graphics window for the true GRASS colors to be displayed on the graphics window. The mouse doesn't have to be clicked inside the window; just moving the mouse within the window is enough to activate the window. While the mouse is outside the GRASS graphics window, the colors in the graphics window will be random.

NOTES

When using the GRASS program *d.colors* or the interact selection in the *d.display* program, the mouse must be within the GRASS graphics window when toggling or highlighting categories. If the mouse is not within the graphics window, the colors will not represent the true category colors. Using the two programs mentioned above while running GRASS from an Intergraph window (ascii window) creates a problem; The mouse must be within the graphics window at all times for those programs to work properly, but the mouse also must be moved to the ascii window to activate the ascii window so that commands can be typed. This is not a problem when GRASS is run from an ascii terminal, because the mouse can be left within the GRASS graphics window at all times.

AUTHOR

Michael Higgins, U.S. Army Construction Engineering Research Laboratory

d.label

NAME
d.label **-** Creates and displays text labels in the active display frame on the graphics monitor.
(GRASS Display Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
d.label
d.label help
d.label [size=value] [backcolor=name] [textcolor=name] [font=name]

DESCRIPTION
d.label allows the user to create and display text labels in the active frame on the graphics monitor. It interactively requests that the user type in text and use the pointing device (mouse) to identify where this text is to be placed within the active graphics frame. The program will prompt the user for label size, background color, text color, and text font type, if the user fails to specify these values on the command line. Program prompts use the standard GRASS *parser* interface described in the manual entry for *parser*.

Parameters:
size=value Sets the label text size to the specified number. Values are stated as a percentage of the frame height; e.g., a size of 10 will make each line of text equal to one-tenth the height of the display frame.
Options: 0-1000
Default: 10

backcolor=name Sets the color of the label background to the name stated.
Options: red, orange, yellow, green, blue, indigo, violet, gray, white, black
Default: black

textcolor=name Sets the color of the label text to the name stated.
Options: same as for backcolor.
Default: white

font=name Sets the font type used for the label's text to name.
Options: romand, romanp, romant, romans, scriptc, scripts, romancs, italicc, italiccs, gothitt, gothgrt, and gothgbt. The user can view these available fonts by running *show.fonts.sh.*
Default: romans (Roman simplex font type), or whatever font the user has set the font type to before entering *d.label*.

NOTES
This program will allow the user to display labels anywhere in the active display frame, even in areas lying outside of the current geographic region.

d.label will only allow the user to type in one line of text. The fully interactive program *d.labels* can be used to create text labels having as many as four lines.

d.label does not create paint labels files displayable by *d.paint.labels* **--** *d.labels* does. These programs should be integrated in future GRASS releases.

BUGS

If the user is running GRASS under X Windows, and the user enters a text label longer than 80 characters (i.e., allows text to wrap onto a second line), X Windows will stop the user's GRASS graphics monitor without throwing the user out of GRASS. If this happens, the user should simply restart a graphics monitor, using *d.mon*.

SEE ALSO

d.display, d.font, d.labels, d.legend, d.mon, d.paint.labels, d.rast, d.scale, d.text, d.title, d.where, show.fonts.sh, parser

AUTHOR

James Westervelt, U.S. Army Construction Engineering Research Laboratory

d.labels

NAME
d.labels **-** To create/edit GRASS paint label files for display on the graphics monitor.
(GRASS Display Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
d.labels

DESCRIPTION
The *d.labels* program allows the user to interactively create and modify paint-labels files for use with the *p.map* and *d.paint.labels* programs. The paint labels files created will be stored in the user's current mapset under the /paint/labels directory.

d.labels is fully interactive, and requires use of a graphics monitor. After the user has typed

d.labels

and has typed in the name of a new or existing paint-labels file, the user is asked to indicate label positions on the display frame using the mouse-pointer, and to edit label parameters that define the label's size, color, and other characteristics through use of a visual-ask (VASK) data entry screen. The mouse can also be used to indicate the geographic coordinates of locations within the current geographic region in the active frame. It is helpful to display a relevant data file in the active display frame before running *d.labels* to provide visual clues for label placement.

USER PROMPTS
You will first be prompted to enter the name of either an old paint-labels file that you wish to modify, or a new paint-labels file that you wish to create. The following is the file name prompt:

```
OPTION: Label text file name

key: name
required: YES
enter option >
```

If you enter the name of a new (non-existent) paint-labels file, you are asked to indicate a location falling within the current geographic region in the active display frame at which the label is to be placed.

Mouse Buttons
 Left: Place Label Here
 Middle: Where Am I ?
 Right: Quit

Once you have indicated a position, you will be presented with a data-entry screen (also known as a VASK screen) so that you can enter the paint-label parameters that define how the label will appear on the display or on the color printer. Once you finish entering the label-parameters, and you hit the <ESC> key to proceed, the label will be displayed on the screen. If you like the appearance of the label, enter "y" to the "Look OK?" prompt; if not, enter "n" and you will again be presented with the data entry screen. Once you approve the label's appearance you will be asked to indicate the position of the next label.

If you choose to display an already-existing paint-labels file, you will be shown the labels. You will then have the option of editing a label or adding more labels.

Buttons
 Left: Edit Existing Labels
 Middle: Quit
 Right: Add more Labels

On-line help is available. If you are puzzled about what should go in a data entry field, just enter "help" in that field, hit the <ESC> key, and you will be presented with a short description of the field and what you may enter.

The following is a list of the different fields in the data entry screen and a short description of each.

TEXT: The user can enter up to four lines of text.

SKIP: yes/no. If no, label will be printed. If yes, the label will be retained in the file but not printed.

LOCATION: Determines where the text will be located. The user specifies the easting and northing in map units, and (optionally) specifies a vertical and horizontal offset (in printer pixels) from the specified easting/northing. (The vertical (y) offset will shift the location to the south if positive, north if negative. The horizontal (x) offset will shift the location east if positive, west if negative.) These offsets are provided to allow finer placement of labels.

RESET LOCATION: yes/no. If yes, the user will be allowed to use the mouse to select a new location for the label. The mouse buttons will function as follows:

 Mouse Buttons
 Left: Place Label Here
 Middle: Where Am I ?
 Right: Quit

PLACEMENT: Determines which part of the label to which the location refers. This may be specified as:

lower left   (lower left corner of the text)
lower right   (lower right corner of the text)
lower center   (bottom center of the text)
upper left(upper left corner of the text)
upper right   (upper right corner of the text)
upper center   (top center of the text)
center   (center of the text)

TEXT SIZE: Determines the size of the lettering used in the label. Text size (height) is stated in units of (ground) meters; e.g., a text size of 500 sets the height of each line of text equal to 500 ground meters. Thus, text will appear to grow or shrink when displayed, depending on the user's current geographic region settings.

TEXT COLOR: This selects the text color. This color can be specified in one of three ways:

1) By color name:
 aqua, black, blue, brown, cyan, gray, green, grey, indigo, magenta, orange, purple, red ,violet, white, yellow

2) As red, green, blue percentages. For example: .5 .4   .7

(This form is not supported by *d.paint.labels*, but see *p.map* and *p.labels*.)

3) By printer color number, to get the exact printer color.
(This form is not supported by *d.paint.labels*, but see *p.map* and *p.labels.)*

Note. If the user sets the text color equal to the background color, the text will not appear on the graphics monitor.

WIDTH: This determines the line thickness of the letters. The normal text width should be set to 1. Larger numbers can be used to simulate bold face. (*d.paint.labels* ignores this value and always uses 1.)

HIGHLIGHT COLOR: The text can be highlighted in another color so that it appears to be in two colors. The text is drawn first in this color at a wider line width, and then redrawn in the text color at the regular line width. Highlight colors are not shown on the graphics display, they are only shown in the output from *p.labels* and *p.map*.

HIGHLIGHT WIDTH: Specifies how far from the text lines (in pixels) the highlight color should extend. Highlight colors are not shown on the graphics display, they are only shown in the output from *p.labels* and *p.map*.

BACKGROUND COLOR: Text may be boxed in a solid color by specifying a background color (see TEXT COLOR above to specify a color). Specify none to use no background.

OPAQUE TO VECTORS: yes/no. This field only has meaning if a background color is selected. yes will prevent vector lines from entering the background. no will allow vector lines to enter the background.

BORDER COLOR: Select a color for the border around the background. Specify none to suppress the border.

NOTES
This program is fully interactive and requires no command line arguments.

If the user types the word help in a data entry field while entering label information, acceptable field choices will be listed.

Labels must be placed within the current geographic region. If the user attempts to place a label outside of this region, the program will appear to loop endlessly. It is simply failing to display a label positioned beyond the boundaries of the current region; in this event, the user should simply reposition the label.

Labels created by this program are stored in ASCII files under the directory $LOCATION/paint/labels.

SEE ALSO
d.label, d.paint.labels, d.rast, d.vect, d.where, p.labels, p.map

AUTHOR
Dave Johnson
DBA Systems, Inc.
10560 Arrowhead Drive
Fairfax, Virginia 22030

NOTICE
This program is part of the alpha section of the GRASS distribution. Unlike the code in the main section of GRASS, the alpha code has not yet been fully tested for one release cycle.

d.leg.thin

NAME
d.leg.thin **-** improved version of *d.legend* that allows:
- Thinning the categories to be represented in the legend
- Displaying a continuous gradient of all categories in the legend
- Interactive mouse placement of the smooth gradient box

GRASS VERSION
4.x, 5.x

SYNOPSIS
d.leg.thin [-nsm] map=name [color=name] [lines=value] [thin=value]

OPTIONS
Flags:
-*n* Do not show category values

-*s* Draw smooth gradient

-*m* Use mouse to size & place legend (only when -s is specified)

Parameters:
map Name of raster map.

color Sets the legend's text color.
Options: red, orange, yellow, green, blue, indigo, white, black, brown, magenta, aqua, gray, grey
Default: white

lines Number of text lines (useful for truncating long legends)
Options: 0-1000
Default: 0

thin Thinning factor (thin=10 gives cats 0,10,20...)
Options: 0-1000
Default: 1

NOTE
Note that when using the mouse to size & place the smooth legend, if you make a horizontal box the values run from left to right and no category values are printed (you may want to use *d.label* for that).

AUTHOR
Bill Brown, U.S. Army Construction Engineering Research Laboratories

d.legend

NAME
d.legend **-** Displays a legend for a raster map layer in the active frame on the graphics monitor.
(GRASS Display Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
d.legend
d.legend help
d.legend map=name [color=name] [lines=value]

DESCRIPTION
d.legend displays a legend for a user-specified raster map layer in the active frame on the graphics monitor. The legend's size is based on the height of the active frame. The user should therefore take care to create a display frame of suitable dimensions before running *d.legend*. *d.legend* will only obscure those portions of the active frame that directly underlie the legend.

The user can run *d.legend* either non-interactively or interactively. If the user specifies the name of a raster map layer on the command line, the program will run non-interactively. Default legend text color and number of lines will be used unless the user specifies other values on the command line. Alternately, the user can simply type *d.legend* on the command line; in this case, the program will prompt the user for parameter values using the standard GRASS *parser* interface described in the manual entry for *parser*.

OPTIONS
Parameters:
map=name Name of a raster map layer whose legend is to be displayed in the active display frame.

color=name Sets the legend text color to the name stated.
Options: red, orange, yellow, green, blue, indigo, violet, magenta, brown, gray, white, and black
Default: white

lines=value Number of lines to appear in the map legend. The number of lines refers to the maximum number of lines of type that can be displayed given the height of the active display frame. If unspecified by the user, the program will divide the display frame into the number of lines required to display all of the category labels and colors associated with the named map. To decrease the size of the text, increase the number of lines.
Options: 1 - 1000
Default: Set lines value equal to the number of map categories

NOTES
The legend text size is based on the number of lines requested (or, by default, on the number of lines needed to display the legend). If the user attempts to display a very long legend in a relatively short display frame, the legend may appear in unreadable small text.

SEE ALSO
d.colormode, d.colors, d.colortable, d.display, d.erase, d.font, d.frame, d.grid, d.label, d.labels, d.rast, d.scale, parser

AUTHOR
James Westervelt, U.S. Army Construction Engineering Research Laboratory

d.mapgraph

NAME
d.mapgraph **-** Generates and displays simple graphics on map layers drawn in the active graphics monitor display frame.
(GRASS Display Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
d.mapgraph
d.mapgraph help
d.mapgraph [input=name] [color=name] [vsize=value] [hsize=value]

DESCRIPTION
d.mapgraph draws graphics that are described in standard input (default) or the UNIX input file name. If commands are entered via standard input, a *ctrl-d* is used to signal the end of input to *d.mapgraph*. This program performs essentially the same function as *d.graph*; however, point locations are specified to *d.mapgraph* in the geographic coordinate system of the user's current mapset and location (i.e., in map coordinates), rather than in graphics display screen coordinates.

The program can be run interactively or non-interactively. The user can run the program completely non-interactively by specifying the name of a file containing *d.mapgraph* graphics commands and the values of all needed parameters on the command line. The user can instead elect to run the program partially interactively, by specifying any/all of the parameters except the graphics *input=name* parameter on the command line. In this case, *d.mapgraph* will expect the user to input *d.mapgraph* graphics commands from standard input (i.e., the keyboard) and will (silently) prompt the user for these graphics commands. Alternately, the user can simply type *d.mapgraph* on the command line, and be prompted for the values of all parameters (the user can still input graphics commands from an input file using this form.) In this case, the user is presented with the standard GRASS *parser* interface described in the manual entry for *parser*.

OPTIONS
Parameters:
input=name Name of a UNIX file containing graphics instructions. Specify the full path name of the file if not in the current directory. If no file name is given, commands are taken from standard input.

color=name Starting color desired for graphics.
Options: red, orange, yellow, green, blue, indigo, violet, magenta, brown, gray, white, and black
Default: white

d.mapgraph is used for drawing simple graphics on top of map layers. The coordinate system used by *d.mapgraph* is the same as that of the map layer displayed in the active display frame on the graphics monitor (or that of the user's current region, if no map is displayed).

The graphics language is simple and uses the following commands:

# comment A line of comment which is ignored in the processing.

move xpos ypos The current location is updated to *xpos ypos* (where these, respectively, are the easting and northing of geographic coordinates stated in the map coordinate system of the user's current GRASS location, falling within the current region and active frame). If unspecified by the user, the current

location becomes (0,0). If, as most likely, the point (0,0) falls outside of the user's current region, graphics drawn there will not appear in the graphics frame.
Note: use *g.region* to obtain the coordinates of current location. Use *d.where* to obtain specific map coordinates of various points on the raster map displayed in the active frame.
Note: there must be a space between *xpos* and *ypos.*

draw xpos ypos A line is drawn in the current color from the current location to the new location *xpos ypos*, which then becomes the current location. *xpos* and *ypos* are (respectively) an easting and northing stated in the map coordinate system of the user's current GRASS location, and located within the user's current geographic region and active frame.
Note: there must be a space between *xpos* and *ypos*.

color color Sets the current color to that stated.
Color options are: red, orange, yellow, green, blue, indigo, violet, magenta, brown, gray, white, black

size xper yper Subsequent text will be drawn such that the text is *xper* percent of the display frame's width and *yper* percent of the display frame height. If not specified by the user, the text size becomes 5 percent of the active frame's width and 5 percent of the frame's height. This is equivalent to entering size 5 5. text line-of-text The stated text is drawn at the current location using the current color and the current size.

icon type size x y Draws an icon of types o, x, or + with specified size at location x,y.
Note: type o designates a square.

polygon
xpos ypos
xpos ypos
.
.
.
The map coordinates appearing on lines beneath the word polygon, one pair per line, circumscribe a polygon which is to be filled with the current color.

NOTES
d.mapgraph is identical to the *d.graph* command, except for the difference in coordinate systems used. *d.mapgraph* will complain if the user enters something to standard input that it does not understand. Blank lines in the input file will result in this error message.

SEE ALSO
d.INTRO, d.frame, d.graph, d.rast, d.zoom, g.region, grass.logo.sh, parser

AUTHOR
James Westervelt, U.S. Army Construction Engineering Research Laboratory

d.measure

NAME

d.measure **-** Measures the lengths and areas of features drawn in the active frame on the graphics monitor. (GRASS Display Program)

GRASS VERSION

4.x, 5.x

SYNOPSIS

d.measure
d.measure help
d.measure [c1=name] [c2=name]

DESCRIPTION

d.measure provides the user with an interactive way to measure the lengths and areas of lines and polygons drawn by the user in the active frame on the graphics monitor. Lines and polygons are drawn using a pointing device (mouse). Each line segment is drawn in colors *c1* and *c2*. A mouse button menu indicates the user can find the geographic coordinates of the cursor location, draw line segments between user-specified vertices, and quit *d.measure*. Lines and polygons are defined by a series of vertices marked by the user. If more than two successive vertices are drawn, *d.measure* prints the area encompassed (*d.measure* will assume the area is closed even if the user has not joined the first and last vertices). Line lengths are stated in the same units as those of the current LOCATION. Areas are stated in hectares, square miles, and square meters.

Lines and polygons drawn using *d.measure* will overlay (not overwrite) whatever display currently appears in the active frame on the graphics monitor. The user can, for example, run *d.rast* or *d.vect* prior to running *d.measure*, and use raster and/or vector maps as a backdrop.

OPTIONS

The user can specify the colors in which line segments will be drawn by setting the values of *c1* and *c2* on the command line. Default line colors (see below) will be used if the user does not specify the values of *c1* and *c2* on the command line.

Parameters:

c1=name The first color in which each line segment is drawn, while being positioned.
  Options: red, orange, yellow, green, blue, indigo, violet, magenta, brown, gray, white, and black
  Default: white

c2=name The second color in which each line segment is drawn, after its vertices are fixed.
  Options: Same as *c1*
  Default: white

NOTES

d.measure uses all segments drawn by the user when computing area. If the user draws an area within another area, the combined area of both regions will be output.

SEE ALSO

d.display, d.frame, d.graph, d.mapgraph, d.rast, d.vect ,d.where

AUTHOR

Jim Westervelt, U.S. Army Construction Engineering Research Laboratory
Michael Shapiro, U.S. Army Construction Engineering Research Laboratory, 4.1 Release

d.menu

NAME
d.menu **-** Creates and displays a menu within the active frame on the graphics monitor.
(GRASS Display Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
d.menu
d.menu help
d.menu [bcolor=name] [tcolor=name] [dcolor=name] [size=value]

DESCRIPTION
d.menu allows the user to create a menu containing a title and options, and to display this menu in the active frame on the graphics monitor. After the menu is displayed in the active frame, the pointing device (mouse) is activated and must be used to select one of the menu options. The number associated with the selected menu option is then printed to standard output and the program exits. This program provides GRASS macro writers with a mouse interface for users.

Parameters can be stated on the command line, from within standard input, or (as illustrated in EXAMPLE) from within a script file.

The user can specify the menu's background, text, and line colors (bcolor, tcolor, and dcolor) and the menu size (size) on the command line. If the user sets at least one of these values on the command line, any remaining values that are not specified on the command line will be set (automatically) to their default values (see below). Alternately, the user can simply type *d.menu* on the command line; in this case, the program will prompt the user for the menu colors and text size using the standard GRASS *parser* interface described in the manual entry for *parser*. The user will then be (silently) prompted to enter the menu's location and contents through standard input (see Menu Information, below).

Parameters:
bcolor=name Sets the color of the menu background (the rectangle on which the text sits).
Options: red, orange, yellow, green, blue, indigo, violet, magenta, brown, gray, white, black
Default: black

tcolor=name Sets the color of the menu text.
Options: Same as those for bcolor.
Default: white

dcolor=name Sets the line color used to divide text lines in the menu.
Options: Same as those for bcolor.
Default: white

size=value Sets the menu's text size to the specified value. Sizes are stated as percentages of the height of the active graphics display frame.
Options: 1 **-** 100
Default: 3 (i.e., 3%)

Menu Information:
After the user has (optionally) specified menu colors and size, the program expects the user to enter information about the menu's location and content. The menu will be placed in the lower left corner of the active display frame by default if the user does not position it elsewhere using the *.T* or *.L* commands. The user specifies the menu contents by typing in a menu title followed by the option choices to appear in the menu when displayed. The user must enter a menu title and at least one menu option.

.T value Specifies the menu's distance from the active display frame's top edge (as a percentage of the active frame's height).
Note: Not required

.L value Specifies the menu's distance from the active display frame's left edge (as a percentage of the active frame's width).
Note: Not required

menu title A title that describes the type of options listed in the menu, and that will appear at the top of the menu when it is displayed.

option name(s) The options that will appear in the menu when displayed. Each menu option should appear on a separate line. The user may enter as many options as desired, but must enter at least one menu option.
Note: The user should choose a menu size and location that will allow all menu options to be displayed in the active frame.

If the user enters the menu title and option(s) from standard input (i.e., at the keyboard rather than from a file), the user should enter control-d to end input and display the menu in the active frame on the graphics monitor. (Note: The *d.menu* program can also be incorporated into UNIX Bourne shell script macros. The below example shows how this might be done.)

EXAMPLE
In the following example, the shell script menu2 calls the shell script color.select which contains *d.menu* commands to display a menu in the current frame on the graphics monitor. After the user selects an option from the display menu, the selection number is available for use by menu2.

Contents of file menu2:

```
 #! /bin/csh -f
set option = 0
set colors = (red green blue black white )
@ option = `color.select`

if ($option <= 5) then
set color = $colors[$option]
echo $color

endif

exit
```

Contents of file color.select:

```
 #! /bin/csh -f
d.menu bcolor=red tcolor=green dcolor=yellow size=5 << EOF
 .T 25
 .L 25
 Color Choices
 Option 1
```

```
          Option 2
          Option 3
          Option 4
          Option 5
          EOF
```

If the user runs menu2, a menu will be displayed on the graphics monitor that has red background, green text, with menu options divided by yellow lines, and a text size of 5% of the active display frame height. The mouse cursor will become active, allowing the user to select (by pointing with the mouse) one of the displayed menu options. Here, these menu options are called Option 1, Option 2, and Option 3, etc. The first line of text (here, the words Color Choices) contains the title of the menu; this line is not a menu option that can be chosen by the user with the mouse. When the user presses one of the mouse buttons while pointing to the desired menu choice, the number of the option chosen will be available for capture by the shell script menu2. menu2 is a simple example that takes this information and only echoes it to the screen.

NOTES

Although the user can vary text size, all text within the same menu is displayed in a single text size (and font). If the user specifies that items included in the menu's text be displayed in different sizes, all text will be displayed in the size stated last.

SEE ALSO

d.display, d.font, d.frame, d.grid, d.label, d.legend, d.paint.labels, d.text, d.title, show.fonts.sh, parser

AUTHOR

James Westervelt, U.S. Army Construction Engineering Research Laboratory

d.mon

NAME
d.mon **-** To establish and control use of a graphics display monitor.
(GRASS Display Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
d.mon
d.mon help
d.mon [-lLprs] [start=name] [stop=name] [select=name] [unlock=name][nlev=value]

DESCRIPTION
The GRASS program *d.mon* allows the user to start, select, list, query the status of, release control of, stop, and unlock control of, available graphics monitors. The user can run this program either interactively (through a series of menus), or non-interactively by typing the name of the monitor to start, stop, select, or unlock, and any desired flags on the command line.

Flags:

-*l*	List all monitors
-*L*	List all monitors (with current status)
-*p*	Print name of currently selected monitor
-*r*	Release currently selected monitor
-*s*	Do not automatically select when starting

Parameters:

start=name	Name of graphics monitor to start
stop=name	Name of graphics monitor to stop
select=name	Name of graphics monitor to select
unlock=name	Name of graphics monitor to unlock
nlev=name	Number of color levels for each R/G/B

When the user enters the command *d.mon* without specifying parameter values, the below menu appears on the screen:

```
MONITOR MENU
Making sure that the graphics monitor is running:

  1 - Start a graphics monitor  (also automatically selects this monitor)
  2 - Stop a graphics monitor

Choosing a graphics monitor for your graphics:

  3 - Select a graphics device for output  (currently selected monitor:  name)
  4 - Release control of the graphics driver  (let someone else use it)
```

```
            (option 4 appears only after selection of a monitor)

        RETURN   quit
```

These parameters perform the following functions:

1   Start a Monitor.
In order to display on-screen GRASS graphics, the user must start and select a graphics monitor. By default, the start command actually runs two commands, to both start and select whatever monitor is named by the user. (The user can get a list of available monitors by setting the *-l* or *-L* flag on the command line.)

When a monitor is started, it is therefore also (automatically) selected for output, unless the *-s* flag is set by the user; the user can also explicitly select a monitor that has been started (see (3) below). After a monitor is started, a blank graphics frame should appear on whatever terminal the user is using to display graphics.

The desired monitor should be started once and need not be restarted unless it is stopped (option 2) for some reason. A monitor may continue to run for any length of time, even when no GRASS session is being run. The monitor program runs in the background.

Using the *nlev* parameter the user can define the number of color levels to be used. *nlev* can be a value between 1 and 256. *nlev*=256 means 8bit for each R, G, B color channel equal to 24bit truecolor (16.7 million colors). The calculation of number of colors (ncolors) is performed by calculating:

```
        ncolors = nlev ^ 3
```

Starting a monitor with a high number of colors may take some time to allocate the required memory.

2   Stop a Monitor.
Sometimes the monitor program needs to be stopped (terminated). Choosing option 2 will terminate a user-specified monitor program.

A graphics monitor has two different types of status: monitor program not running, and monitor running. A monitor that has been started and/or selected will be listed as running; a monitor that has been stopped (or not started) will be listed as not running. The *-L* (list status) flag will list the status of each monitor connected to the system.

3   Select a Monitor.
When the user starts a monitor, it is also (automatically) selected for graphics output unless the user sets the *-s* flag. In order to use (direct graphics output to) a monitor, the user must select that monitor for use, either by simply starting the monitor without the *-s* flag or by explicitly selecting the monitor for output using option 3. Only running monitors can be selected for graphics output. Once the user has selected a monitor for output, no other user can use this monitor for graphics output until the monitor driver is either released (by the user) or unlocked (by any user on the system). The user can run multiple graphics monitors by simply starting each of the graphics monitors drivers he wishes to direct output to.

4   Release a Monitor.
Once a user has selected a monitor for graphics output, it is locked for use by that user until either: (1) the user voluntarily releases control of the monitor for use by another (option 4), or (2) another GRASS user unlocks the user's control of the monitor. Menu option 4 appears only to the person who has selected the monitor (since only that user can release control of his selected graphics monitor.) If another user wishes to unlock the user's control of the monitor, that user must run *d.mon* from the command line and set the *unlock=name* parameter.

You may choose multiple options within the *d.mon* program. To leave (exit) the *d.mon* menu, press the <RETURN> key.

NOTES

The *d.mon* program can regulate control of graphics monitors both in systems using multiple monitors and in systems using a single graphics monitor.

SEE ALSO

d.erase, d.frame, monitorcap

AUTHOR

Michael Shapiro, U.S. Army Construction Engineering Research Laboratory,
Full 24bit support by Roberto Flor, ITC-Irst, Trento, Italy

d.paint.labels

NAME

d.paint.labels **-** Displays text labels formatted for use with GRASS paint (*p.labels, p.map*) output to the active frame on the graphics monitor.
(GRASS Display Program)

GRASS VERSION

4.x, 5.x

SYNOPSIS

d.paint.labels
d.paint.labels help
d.paint.labels file=name

DESCRIPTION

d.paint.labels displays the paint label file name in the active display frame on the graphics monitor. This file can be generated by the labels option in the *p.labels* program, the *d.labels* program, or simply created by the user as an ASCII file (using a text editor like vi) and placed in the appropriate /paint/labels directory under the user's current mapset and location (i.e., under $LOCATION/paint/labels).

OPTIONS

Parameter:

file=name	Name of ASCII file containing paint labels.

This program can be run either non-interactively or interactively. If the user specifies the name of a paint labels file on the command line, the program will run without interacting with the user. Alternately, the user can type simply *d.paint.labels* on the command line; in this case, the program will prompt the user for the name of a paint labels file using the standard GRASS *parser* interface described in the manual entry for *parser*.

NOTES

Some of the information stored in the label file is unused by *d.paint.labels*. The unused information found in a label file is printed to standard output. This extra information is used by such programs as *p.labels* and *p.map.*

SEE ALSO

d.display, d.font, d.label, d.labels, d.text, d.title, p.labels, p.map, parser

AUTHOR

James Westervelt, U.S. Army Construction Engineering Research Laboratory

d.param.scale

NAME

d.param.scale **-** GRASS module that allows interactive identification of terrain parameters and features with a mouse. Reports membership classes for variable window sizes and their variance or entropy.

GRASS VERSION

4.x

SYNOPSIS

d.param.scale [-cft] in=name [s_tol=value] [c_tol=value] [size=value] [param=name] [exp=value] [zscale=value]

OPTIONS

Flags:

-*c* Constrain model through central window cell

-*f* Fix axis scale

-*t* Create text output of results

Parameters:

in Raster surface layer to interrogate

s_tol Slope tolerance that defines a `flat' surface (degrees)
Default: 1.0

c_tol Curvature tolerance that defines `planar' surface
Default: 1.0

size Size of processing window (odd number only)
Default: 15

param Morphometric parameter to calculate
Options: elev, slope, aspect, profc, planc, longc, crosc, minic, maxic, feature, resid, moran, geary
Default: elev

exp Exponent for distance weighting (0.0-4.0)
Default: 0.0

zscale Vertical scaling factor
Default: 1.0

***d.param.scale* can calculate the following:**

elev: Generalized elevation value.

slope: Maximum gradient at a point.

aspect: Direction of maximum gradient.

profc: Profile convexity (vertical in direction of steepest slope).

planc: Plan convexity (contour curvature).

crosc: Cross sectional convexity (tangent to contours, downslope).

longc: Longitudinal convexity (perpendicular to contours downslope).

minic: Minimum convexity.

maxic: Maximum convexity.

features: Morphometric features:
peaks, ridges, passes, channels, pits and planes.

SEE ALSO
r.param.scale

d.points

NAME
d.points **-** Displays point graphics in the active frame on the graphics display monitor.
(GRASS Display Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
d.points
d.points help
d.points [color=name] [size=value] [type=name] [file=name]

DESCRIPTION
d.points displays points whose coordinate locations are specified by the user and located within the current geographic region in the active display frame on the graphics monitor. Point coordinates are read either from standard input or from a file stated by the *file=name* option. When entering coordinates through standard input, the user presses *control-d* to signal the end of input to *d.points.*

The program can be run interactively or non-interactively. The user can run the program completely non-interactively by specifying the name of a points file (containing the geographic coordinates of icons to be sited) and the values of all needed parameters on the command line. If the name of an input file to *d.points* is given on the command line, any other parameter values left unspecified on the command line (e.g., the icon color, size, and/or type) are set to their default values.

The user can instead elect to run the program partially interactively, by specifying any/all of the parameters except the points *file=name* parameter on the command line. In this case, *d.points* will expect the user to enter icon locations from standard input (i.e., at the keyboard) and will prompt the user for these locations. Each icon location should be entered as an easting and northing pair separated by a space, followed by a carriage return. Coordinates should be stated in whatever map coordinate system is used by the user's current GRASS location. The user presses *control-d* to signal the end of standard input to *d.points*.

Alternately, the user can simply type *d.points* on the command line, and be prompted for the values of all parameters (the user can still input icon coordinates from an input file using this form). In this case, the user is presented with the standard GRASS *parser* interface described in the manual entry for *parser*.

Parameters:
color=name Sets the current color to the name stated.
Options: red, orange, yellow, green, blue, indigo, violet, gray, white, black
Default: gray

size=value Size of icon in pixels.
Options: 0-1000 (pixels)
Default: 5 (pixels)

type=name Sets the shape (type) of the icon to be drawn to name.
Options: x, diamond, box, +
Default: +

file=name Take input from the UNIX file name.
(Note: This file must be in the user's current directory or given by its full path name.)

Default: Standard input. The file contents should consist of a series of geographic coordinates that fit within the boundaries of the current geographic region falling within the active display frame (refer to *d.where* or *g.region* for these coordinates). Each icon location should be specified by an easting and northing (in that order) separated by a single blank space, and followed by a carriage return. Eastings and northings should be stated in whatever map coordinate system is being used by the user's current GRASS location.

NOTES

d.points will display a point at the location specified in the active frame, even if the point lies outside the boundaries of current geographic region.

SEE ALSO

d.frame, d.where, d.sites, g.region, s.out.ascii, parser

AUTHOR

James Westervelt, U.S. Army Construction Engineering Research Laboratory

d.profile

NAME
d.profile **-** Displays profiles of a user-specified raster map layer.
(GRASS Display Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
d.profile

DESCRIPTION
This command works only interactively. It clears the entire graphics screen and provides a graphical interaction allowing the selection of transects for which profiles are then generated.

USER PROMPTS
First, you will be presented with a prompt asking you to choose a raster map layer to be profiled. Once you specify a valid raster map layer name, the map layer will be displayed in the left half of the graphics display, and the right half of the display will be divided into four display frames. There will also be two frames along the top of the display: a mouse-button menu frame on the left, and a status frame on the right.

The mouse-button menu initially offers you three options:

```
Mouse Buttons:
   Left:   Where am I?
   Middle: Mark FIRST Point of Profile Line.
   Right:  QUIT this.
```

You may query the displayed raster map layer by indicating points with the left mouse-button. The coordinates and category value of each point that you indicate will be displayed on in the status frame. If you mark the first point of the profile line you will be presented with the following mouse-button menu:

```
Mouse Buttons:
   Left:   Where am I?
   Middle: Mark SECOND Point of Profile Line.
   Right:  QUIT this.
```

Once you mark the second point of the profile line, the profile line will be labeled (with a letter from A to D) and displayed in one of the four display frames on the right hand side of the screen. You will then be presented with a third mouse-button menu:

```
Mouse Buttons:
   Left:   DO ANOTHER
   Middle: CLEAR DISPLAY
   Right:  QUIT this.
```

If you would like to view another profile, click on the left mouse-button. If you would like to redisplay the raster map layer and clear out the four profile frames, click on the middle mouse-button. If you would like to quit, then click on the right button.

AUTHOR
Dave Johnson, DBA Systems, Inc.

d.rast

NAME

d.rast **-** Displays and overlays raster map layers in the active display frame on the graphics monitor. (GRASS Display Program)

GRASS VERSION

4.x, 5.x

SYNOPSIS

d.rast
d.rast help
d.rast [-oi] map=name[catlist=list][vallist=list][bg=color]

DESCRIPTION

d.rast displays the raster map layer name in the active display frame on the graphics monitor.

The program can be run either non-interactively or interactively. If the user specifies the name of a raster map layer (map=name) and (optionally) the -*o* option on the command line, the program will run non-interactively. Alternately, the user can type simply *d.rast* on the command line; in this case, the program will prompt the user for the flag setting and parameter value using the standard GRASS *parser* interface described in the manual entry for *parser*.

OPTIONS

Flags:

-*o* Overlay the named raster map layer onto whatever is already displayed in the active graphics frame. Any zero category value data areas in the named raster map will seem transparent, and reveal the underlying image previously displayed in the graphics frame. If the -*o* flag is set, only cells containing non-zero category values will be displayed from the overlaid raster map. All other areas (i.e., the portions of the overlaid map that contain category value 0) will leave the underlying display untouched. If the -*o* flag is not set by the user, *d.rast* will (by default) completely overwrite whatever appears in the active graphics display frame.

-*i* Invert catlist.

Parameters:

map=name Name of the raster map to be displayed. If the active graphics frame already contains text or graphics, and the user does not wish to use the -*o* option, it is wise to first invoke *d.erase* to clear the active graphics frame before running *d.rast*. After running *d.rast*, other *d*. programs like *d.vect* and *d.grid* can be used to enhance the plot.

catlist=list List of categories to be displayed.

vallist=list List of values to be displayed.

bg=color Background color for null.

NOTES

The *d.rast* raster map overlay option (-*o*) only works when the color look-up table for the graphics monitor is set to *d.colormode* fixed.

SEE ALSO

d.choose, d.colormode, d.colors, d.colortable, d.display, d.erase, d.grid, d.overlay, d.vect, d.what.rast, r.reclass, *parser*

AUTHOR

James Westervelt, U.S. Army Construction Engineering Research Laboratory

d.rast.arrow

NAME
d.rast.arrow **-** Draws arrows representing cell aspect direction for a raster map layer.
(GRASS Display Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
d.rast.arrow
d.rast.arrow help
d.rast.arrow [map=name] [type=name] [arrow_color=name] [grid_color=name] [x_color=name] [unknown_color=name]

DESCRIPTION
d.rast.arrow is designed to help users better visualize surface water flow direction, as indicated in an aspect raster map layer. There are two ways to specify the aspect layer the program is to use. The first is to display the aspect map layer on the graphics monitor before running *d.rast.arrow*. The second method involves setting the map parameter to the name of the desired aspect map. This allows the arrows to be drawn over any other maps already displayed on the graphics monitor.

d.rast.arrow will draw an arrow over each displayed cell to indicate in which direction the cell slopes. An arrow can point in one of eight directions. If the aspect layer has a category value denoting locations of "unknown" aspect, *d.rast.arrow* draws a question mark over the displayed cells of that category. Cells of category 0 (no data) will be marked with an "X".

Note: Arrows are cell size and cannot be seen if the raster map is relatively close in scale. From a command line mode, the user cannot run *d.rast.arrow* and then run *d.rast.zoom* and still see the arrows. The zooming action erases the arrows and the user must run *d.rast.arrow* again.

OPTIONS
Parameters:
map=name Name of an existing raster map layer to be displayed.

type=name Type of existing raster data to be displayed.
Options: grass, agnps, answers
Default: grass
Using this information, *d.rast.arrow* uses internal information to convert category values into appropriate arrow directions.

arrow_color=name Color in which arrows will be drawn.
Options: white, red, orange, yellow, green, blue, indigo, violet, magenta, brown, gray, black
Default: green

grid_color=name Color in which grid outlines will be drawn.
Options: white, red, orange, yellow, green, blue, indigo, violet, magenta, brown, gray, black
Default: gray

x_color=name Color in which x's will be drawn.
Options: white, red, orange, yellow, green, blue, indigo, violet, magenta, brown, gray, black
Default: white

unknown_color=name Color in which unknown information will be displayed.
Options: white, red, orange, yellow, green, blue, indigo, violet, magenta, brown, gray, black
Default: red

SEE ALSO
d.frame, d.rast, d.rast.edit, d.rast.num, d.rast.zoom, d.zoom, g., r.mask, r.slope.aspect, parser

AUTHOR
Chris Rewerts, Agricultural Engineering, Purdue University.

d.rast.edit

NAME
d.rast.edit **-** Program allowing users to interactively edit the cell category values of raster map layers displayed on the graphics monitor.
(GRASS Display Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
d.rast.edit
d.rast.edit help
d.rast.edit [grid_color=name]

DESCRIPTION
The *d.rast.edit* program allows users to interactively edit cell category values in a raster map layer displayed to the graphics monitor using a mouse cursor. This program determines the name of the raster map layer currently displayed in the active frame on the selected graphics monitor (if none, program will abort). The user is then prompted for the name of a new raster layer to be created in the user's mapset. *d.rast.edit* does not modify the user's original raster map layer.

OPTIONS
Parameter:
grid_color=color Sets the color to be used for a grid drawn during edit mode. A well-selected grid color helps to identify individual cells in the output.
Options: red, orange, yellow, green, blue, indigo, violet, gray, white, black
Default: black

Geographic Region Concerns
d.rast.edit reads the region definition for the raster map layer being edited from its cell header file. The new, edited copy of the raster layer will be created with the same resolution and region dimensions. If the current region resolution does not match the raster map layer's resolution, the program will abort. The north, south, east, and west geographic region settings of the current region can be set to any view, as long as they fall within the boundaries of the raster map layer described in its cell header. This is important, since the current view must be such that individual cell locations are easily visible and sufficiently large on the graphics monitor to be pointed at with the mouse cursor. Any mask in place will be ignored when writing the new raster map layer.

Mouse Menus
Cell editing is done using a mouse cursor to choose menu selections and indicate areas on the displayed raster map that are to be edited.

Selections available via the Main Menu and the subsequent functions or sub-menus are described below:

1. Edit
invokes edit mode, during which no graphic menus are used, so that none of the displayed map is hidden. At the start of the edit mode, a grid is drawn over displayed cells to help distinguish cell boundaries. Users can select the color of this grid by setting the *grid_color* parameter. Interaction with the program during edit mode is done by using the three mouse buttons as follows:

Left Button

what's here. Identify the category value of the cell under the mouse cursor when the button is pressed. The category value of this cell is printed in the text frame.

Middle Button
edit. The current category value of the cell under the mouse cursor is iterated and a prompt for a new cell value appears on the textual command frame. (Remember to move the mouse to the text frame). Enter a new cell category value which is within the range of current category values for the map layer. Note: the edited cell is displayed in the color of the newly-assigned category value, but is hatched with lines of the grid color to indicate it has been edited, since, if the value of the cell is polled (before leaving edit mode), the old category value will be reported.

Right Button
exit edit mode. Command is returned to the Main Menu on the graphics monitor. If cell value changes were made during edit mode, they will be saved upon exit.

2. Redraw
redisplays the raster map layer on the graphics monitor.

3. Zoom
calls the *d.rast.zoom* program, to allow changing the view of the raster layer. Zoom's instructions will appear on the text command window.

Interaction is conducted using mouse keys.

4. Arrow
This function is available for users editing aspect maps. The *d.rast.arrow* program is called, which draws arrows over the displayed cells to indicate the downslope direction of the cell. After selecting the arrow selection from the main menu, sub-menus will appear, allowing input options to be set for the *d.rast.arrow* program. The first sub-menu selects the type of aspect data displayed. (regular "GRASS" format, as produced by *r.slope.aspect*; a format prepared as input to the "AGNPS" program; and a format prepared for the "ANSWERS" program. The *d.rast.arrow* program can accept the name of a layer not drawn on the display for use as input, and an option is given to input the name of that layer, if desired. Otherwise, the currently displayed map layer will be used. Next, the color options of *d.rast.arrow* may be set via separate menus, or a choice for using default settings may be used.

5. Number
calls the *d.rast.num* program, which will print the cell category values over the displayed cells on the graphic monitor.

5. Options
change the program options setting for grid_color.

6. Exit
quit the *d.rast.edit* program. If edits have been made, the new raster map layer will be created. Support files are constructed. Category labels and color maps (if any) are copied from the original layer. *d.rast.edit* redisplays the original raster map file on the monitor as it exits.

NOTES
d.rast.edit will not create a new raster map layer if the user makes no cell edits while running the program.

Be careful not to cover the graphics monitor window with another frame during the editing process. In some cases, the redraw option will be able to remove traces of other frames; otherwise, the graphics monitor will not be refreshed until after *d.rast.edit* exits. Further note, however, that this has only been tested on SUN workstations, using a modified SUNVIEW graphics monitor driver.

The primary bane of the *d.rast.edit* program involves large map layers (with lots of rows and columns) and/or slow computers, since the program must read and write raster map layers row by row for the full size of the map layer as dictated by its region size and resolution. (The current region settings of north, south, east, and west will not limit the size of the edited copy of the map layer, since by use of the *d.rast.zoom* program, these values may change several times during the editing session). Their effects could be lessened if the program were to create a table of changes that it could incorporate into the new raster file on a forked process started when the user exits, or otherwise allow the user to issue a "save" command when he has made all desired edits to the raster file. Currently, for instance, if the user needs to use *d.rast.zoom* to access a different area of the map, he must wait for a read and write for each entrance and egress of the edit mode.

There is no "undo" command or way to exit edit mode without saving changes.

It would be nice to incorporate a scrollable version of *d.legend* (such that one could see a legend for files with many categories on a standard size sub-frame). It would be even nicer to be able to select the category values from a graphical legend when editing cell values (thereby saving a trip to the text frame to type in the new value).

Perhaps method(s) for multiple or mass cell edits would be useful. This could be done by providing modes in which the user may: 1) edit a block of cells to a given value by drawing a box; 2) be able to choose a given value which is automatically used as the new value on each cell chosen until a different value is desired.

There is no interrupt handling. This could leave files in .tmp or (rarely) result in half-baked raster files. The original file would survive unscathed by an interrupt at most any point in execution, but the graphics monitor may be left in an indeterminate state (try *d.erase* or *d.mon select=monitor_name* to bring it back into shape). Beware of exiting the program by means other than using exit on the Main Menu.

Perhaps a grid_color option allowing the grid to be turned off is needed, since it may be inappropriately displayed (i.e., not over the cell edges) in some circumstances.

The program has no method to enter new values beyond the current range of categories, but additional programming could make it so.

SEE ALSO
d.erase, d.frame, d.mon, d.rast, d.rast.arrow, d.rast.num, d.rast.zoom, d.zoom, g.region, r.mask, r.slope.aspect

AUTHOR
Chris Rewerts, Agricultural Engineering, Purdue University.

d.rast.num

NAME

d.rast.num **-** Overlays cell category values on a raster map layer displayed to the graphics monitor.
(GRASS Display Program)

GRASS VERSION

4.x, 5.x

SYNOPSIS

d.rast.num
d.rast.num help
d.rast.num [map=name] [grid_color=name]

DESCRIPTION

d.rast.num overlays cell category values onto a raster map layer displayed to the user's graphics monitor. Category values will be displayed in white and/or black, based on the colors in which underlying cells are displayed. A grid outlining each map cell will also be overlain in a user- specified color.

The user should run *d.rast* to display the desired map layer on the graphics monitor before running this program.

d.rast.num can be run non-interactively or interactively. If the user specifies the map whose category values are to be displayed and/or the grid color on the command line, the program will run non-interactively. Alternately, the user may simply type *d.rast.num* on the command line; in this case, the program will prompt the user for parameter values using the standard GRASS user interface described in the manual entry for *parser*. If no grid color is given the default will be used. If no map layer is specified, the program will use whatever raster map layer is currently displayed in the active frame on the graphics monitor.

COMMAND LINE OPTIONS

Parameters:

map=name Name of existing raster map layer whose category values will be displayed.
Default: (whatever raster map layer is currently displayed)

grid_color=name Color in which an overlain grid will be displayed.
Options: white, red, orange, yellow, green, blue, indigo, violet, magenta, brown, gray, black
Default: gray

NOTES

The user is advised to set the current region to a relatively small area (i.e., less than 100 rows by 100 columns); otherwise, the individual cells being displayed will be small and the category value associated with each will be difficult to see.

SEE ALSO

d.erase, d.frame, d.grid, d.mon, d.rast, d.rast.arrow, d.rast.edit, d.rast.num, d.rast.zoom, d.zoom, g.region, r.mask, r.slope.aspect

AUTHORS

Raghavan Srinivasan, and Chris Rewerts, Agricultural Engineering, Purdue University

d.rast.zoom

NAME
d.rast.zoom **-** To interactively zoom in or zoom out of regions on a raster map displayed in the active frame on the graphics monitor.
(GRASS Display Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
d.rast.zoom
d.rast.zoom help

DESCRIPTION
d.rast.zoom allows the user to zoom in or out of a region of interest on the raster map layer currently displayed in the active frame of the graphics monitor. The user zooms in and out interactively, using a mouse to establish the zoom region. To zoom, click the left mouse button to establish a corner, then drag the mouse to establish the second (diagonal) corner and hit the right mouse button to accept the region. "Unzoom" will move back one step (i.e., to the last modified region). If the user tries to unzoom once again, the zoom region will be set equal to the default region settings. The user can zoom or unzoom any number of times until satisfied.

NOTES
d.rast.zoom is sensitive to the current mask (see *r.mask*).

This program runs only interactively.

SEE ALSO
d.frame, d.rast, d.rast.arrow, d.rast.edit ,d.rast.num, d.zoom, g.region, r.mask, parser

AUTHOR
Raghavan Srinivasan, Agricultural Engineering, Purdue University

d.rgb

NAME

d.rgb **-** Displays three user-specified raster map layers as red, green, and blue overlays in the active graphics frame.
(GRASS Display Program)

GRASS VERSION

4.x, 5.x

SYNOPSIS

d.rgb
d.rgb help
d.rgb [red=name] [green=name] [blue=name] [out=name]

DESCRIPTION

RGB stands for red, green, and blue. This program visually overlays up to three raster map layers, each displayed in either red, green, or blue. As each overlay is displayed in a single band, the intensity at each point is the average intensity of the red, green, and blue components of the currently active color table. For example, any cell that carries 100% intensity for either red, green, or blue and 0% intensity for the other two colors will be represented at 33% intensity.

WARNING: Maps that are using the "color wave" color table appear to be solid grey when using *d.rgb*. The average intensity of the colors being used is always 33%.

This program sacrifices spatial resolution to provide full color information. Any color that cannot be fully represented at a particular pixel passes the extra color to the next cell right and down. For example, if a cell is to show 50% red, but the closest color available (without showing more than 50%) is 40%, the pixel to the right and the pixel below will be given an extra 5% red each. In this way all the color is provided at the small cost of a slight amount of blurring. The result is very pleasing, especially on high resolution screens.

COMMAND LINE OPTIONS

Parameters:

red=name	Name of raster map layer to be used for RED component.
green=name	Name of raster map layer to be used for GREEN component.
blue=name	Name of raster map layer to be used for BLUE component.
out=name	Name of raster map layer to contain results.

NOTES

It produces an image and (optionally) a raster map layer with 1000 colors (10 intensities each of red, green and blue). The image and raster map layers will not display properly if the graphics device does not have at least 1000 colors while the user is running in float colormode. Thus, unless the display device has 1000 colors, it is necessary to run the GRASS program *d.colormode mode=fixed* before running *d.rgb*. Otherwise, the colors will be incorrect. The intensity or the saturation layers may be left out. This means that it is possible to have his, hi, or hs images.

SEE ALSO

blend.sh, d.colormode, d.colors, d.colortable, d.his, hsv.rgb.sh, r.mapcalc, rgb.hsv.sh

AUTHOR

James Westervelt, U.S. Army Construction Engineering Research Laboratory

d.rhumbline

NAME
d.rhumbline **-** Displays the rhumbline joining two user-specified points, in the active frame on the user's graphics monitor.
(GRASS Display Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
d.rhumbline
d.rhumbline help
d.rhumbline coor=lon1,lat1,lon2,lat2 [lcolor=name]

DESCRIPTION
A rhumbline is a line following a constant angle of the compass (i.e., a line of constant direction). There are 32 points on the compass (points are roughly 11 degrees 15 minutes apart). *d.rhumbline* displays the rhumbline joining any two user-specified points in the active frame on the user's graphics monitor. The named coordinate locations must fall within the boundaries of the user's current geographic region.

The user can run this program either interactively or non- interactively. If the user simply types *d.rhumbline* on the command line without specifying parameter values, the mouse will be activated and the user will be asked to use the mouse to indicate the two endpoints of the rhumbline. The rhumbline is then drawn in the default color (white). The program also outputs the coordinate locations of the two endpoints on the user's terminal and the number associated with the mouse button depressed by the user in a format useful for input to other programs.

Alternately, the user can specify the starting and ending longitude/latitude coordinates of the rhumbline and (optionally) the color in which the rhumbline will be displayed; in this case, the program will run non-interactively.

OPTIONS
Parameters:
coor=lon1,lat1,lon2,lat2 Starting and ending coordinates of the rhumblines to be displayed, given as longitudes and latitudes.

lcolor=name Rhumbline color.
Options: red, orange, yellow, green, blue, magenta, indigo, violet, gray, white, black
Default: white

NOTES
This program works only with longitude/latitude data bases.

SEE ALSO
d.geodesic

AUTHOR
Michael Shapiro, U.S. Army Construction Engineering Research Laboratory

d.save

NAME
d.save **-** Creates a list of commands for recreating screen graphics.
(GRASS Display Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
d.save
d.save help
d.save [-ca] [frame=name[,name,...]]

DESCRIPTION
When GRASS display (*d.*) commands are used to generate graphical images on the graphics monitor, some commands are noted in lists that the current graphics driver (see manual entry for *d.mon*) maintains. This command, *d.save*, uses this information to create a shell script that can be used to recreate graphics in another frame at another time. The shell script is sent to standard output (the user's terminal). If you wish to save the shell script created by *d.save*, its output should be redirected to a file; for example:

```
d.save > script.out
```

The user can run the program non-interactively by specifying flag settings and parameter values on the command line. If the user types *d.save* without specifying these program arguments, the user will be prompted for inputs through the standard user interface described in the manual entry for *parser*.

OPTIONS
Flags:
-*c* Shell script will contain only information for regenerating graphics from the active (current) frame. Use of this flag overrides the frame option (below).

-*a* Save information needed to regenerate all the frames on the graphics monitor. The shell script can be used to completely reconstruct the contents of the graphics screen.

Parameter:
frame=name[,name,...] Name(s) of those display frame(s) to be saved.
Options: All available frames
Default: Active frame

NOTES
Display commands used interactively, and those that take their graphic instructions from standard input, will not be saved by *d.save*.

SEE ALSO
d.frame, d.mon, d.savescreen, g.region, parser

AUTHOR
David Satnik, Central Washington University

d.scale

NAME
d.scale **-** Overlays a bar scale and north arrow for the current geographic region at a user-defined location in the active display frame.
(GRASS Display Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
d.scale
d.scale help
d.scale [-m] [bcolor=name] [tcolor=name] [at=x,y]

DESCRIPTION
d.scale overlays a bar scale and north arrow for the current geographic region at a user-defined location in the active display frame. The scale and north arrow are proportioned to fit in the active frame when placed along the frame's left edge.

The user can specify the location of the scale and north arrow interactively (using a mouse), or provide the geographic coordinates at which they will be placed. The scale and north arrow will only overwrite those portions of the graphics display that lie directly beneath them.

The user can also specify the colors in which the scale and north arrow will be drawn. By default, if unspecified by the user, a white scale and north arrow will be displayed on a black background.

OPTIONS
The user can run this program either non-interactively or interactively. The program will be run non-interactively if the user specifies program parameter values and desired flag settings on the command line, using the form:

d.scale [-m] [bcolor=name] [tcolor=name] [at=x,y]

If at least one parameter value is given on the command line and the -*m* flag is not set, the program will be run non-interactively; any other parameter values left unspecified on the command line will be set equal to their default values (see below). Alternately, the user can simply type *d.scale* on the command line, without program arguments. In this case, the user will be prompted for needed parameter values and the flag setting using the standard GRASS *parser* interface described in the manual entry for *parser*. If the user sets the -*m* flag, *d.scale* will expect the user to designate the scale's location interactively, using the mouse.

Flag:
-*m* Use the mouse to interactively place the location of the scale and north arrow.

Parameters:
bcolor=name Set the background color underlying the scale and north arrow to name.
Options: red, orange, yellow, green, blue, indigo, violet, gray, brown, magenta, white, black
Default: black

tcolor=name Set the foreground color in which the text, scale, and north arrow are displayed to name.
Options: Same as *bcolor* colors.
Default: white

at=x,y The map easting (x) and northing (y) geographic coordinates designating the location at which the upper left corner of the scale is to be placed. The user should choose geographic coordinates located within the current geographic region (see *d.where* and *g.region* for these coordinates).
Default: 0.0,0.0

NOTES
d.scale assumes that layer units are in meters.

The scale that *d.scale* generates is probably not suitable for very tiny maps (in small display frames).

SEE ALSO
d.legend, d.measure, d.what.rast, d.what.vect, d.where, g.region, parser

AUTHOR
James Westervelt, U.S. Army Construction Engineering Research Laboratory

d.site.edit

NAME
d.site.edit **-** Allows the user to interactively query and update/delete the category contents of single site map layer at user- specified locations within the current geographic region. (GRASS Display Program)

GRASS VERSION
4.x

SYNOPSIS
d.site.edit
d.site.edit help
d.site.edit [input=old_site_name] [output=output_site] [dcolor=color]

DESCRIPTION
d.site.edit outputs the east, north coordinate values and labels associated with site(s) nearest to the user-specified location(s) on user-named site map layer.

The program will query the contents of site map layer named by the user on the command line and allow to either update the description of a site or remove a site. The input site layer must exist in the user's current mapset search path.

The program activates the mouse, and expects the user to indicate the site location(s) to be queried by depressing a mouse button over desired location(s) within the current geographic region in the active display frame on the graphic monitor.

Parameters:

input=old_site_name	Name of existing site map layer.
output=new_site_name	Name of a new site map layer.
dcolor=name	Sets the display color to name.

Options: red, orange, yellow, green, blue, indigo, violet, magenta, brown, gray, white, black
Default: red

d.site.edit allows to edit or display the contents of the geographic coordinates of a location pointed to, and, for the site map layer, and category label nearest site in the named site map layers at this location.

EXAMPLE
It is helpful, but not necessary, to first display a map to be used for reference in the active display frame before running *d.site.edit*. For example, the user might type the following series of commands and receive the output below.

```
d.sites map=wells
```

To first display the soils map in the active frame.

```
d.site.edit input=wells output=new_wells
```

User then moves the mouse to desired location on the displayed well map layer, and presses the middle mouse button to query the category contents of the nearest
well location. The program then outputs the below information to the user's terminal.

```
-613988.990000 east, 804343.040000 north, 08376300 1968 1980 1 3
```

The query may be repeated as often as desired using the left mouse button. The right button on the mouse is used to quit the *d.site.edit* session. The left button will allow get into update menu where using left click the category information can be updated, using middle mouse option a site could be deleted and the right mouse option will bring back to main menu.

NOTES

d.site.edit will always print its output to the terminal screen. *d.site.edit* output can be redirected into a file; however, if it is, the output will go both to the screen and to the file. For example: .nf *d.site.edit* input=wells output=new_wells > what.out .fi will both send *d.site.edit* output to the screen and capture its output in the file named what.out.

SEE ALSO

d.sites, d.what.site, d.what.rast, d.what.vect, d.where, g.region, parser

AUTHOR

Raghavan Srinivasan, Blackland Research Center, TAES, Temple, TX 76502

d.siter

NAME
d.siter **-** Interactively displays subsets of a sites list based on site attributes (GRASS Display Program)

GRASS VERSION
4.x

SYNOPSIS
d.siter
d.siter help
d.siter name

DESCRIPTION
d.siter is a script that calls *s.info* to determine the number, types and ranges of site attributes, then calls a tcl script that presents a user interface. On the interface there are two sliders for each dimension, category, or decimal attribute in the sites file. If a header with labels is in the file, the sliders will be labeled. The value range for each slider pair is determined by the range for the corresponding attribute found in the sites file for sites within the current region. At startup one slider in each pair is set at the minimum value and the other slider is set at the maximum value. The user adjusts sliders as needed to set ranges of desired values for each attribute, dimension, or category. The interface also has controls for setting the size, color, and type of marker, and buttons for display, erase, reset scales, and quit.

- When the display button is chosen, the GRASS program *d.sites.qual* is called with rules derived from the current state of the interface and the specified subset is drawn in the active GRASS monitor. By adjusting the size, color and type of marker and drawing multiple subsets, the user may more easily visualize spatial relationships of various subsets or determine inconsistent attribute values.
- Subsets may be saved by entering a filename in the text entry box of the interface and choosing save as:.
- Due to the limited resolution of sliders and considering the wide range and distribution often found in site attribute data, a rescale button is provided. Choosing rescale results in the currently defined ranges becoming the new min and max and the sliders being reset to their left and right extremes, allowing further refinement of the range.
- To undo the rescale operation, the user may select reset scales that resets all scales to their startup values (loosing any ranges already specified).

OPTIONS
This program can be run non-interactively or interactively. If the user gives the name of a GRASS site_lists map on the command line the program will be run non-interactively. Alternately, the user can type simply *d.siter* on the command line; in this case, the program will prompt the user for a site_lists map.

BUGS
Due to pixel rounding, startup ranges for sliders may be slightly off for floating point attributes. The slider min value will be slightly lower than the min value found in the file and the slider max value will be slightly higher.

SEE ALSO
d.mon, d.sites, d.sites.qual, s.info

AUTHOR
Bill Brown, UI GMS Laboratory

d.sites

NAME

d.sites **-** Displays site markers in the active display frame on the graphics monitor.
(GRASS Display Program)

GRASS VERSION

4.x, 5.x

SYNOPSIS

d.sites
d.sites help
d.sites sitefile=name [color=name] [size=value] [type=name]

DESCRIPTION

d.sites displays a GRASS site_lists map using site markers of a color, size, and type specified by the user. Output is displayed in the active frame on the graphics monitor.

This program can be run non-interactively or interactively. If the user gives the name of a GRASS site_lists map and (optionally) specifies other parameter values on the command line the program will be run non-interactively; any parameter values left unspecified are set to their default values (see below). Alternately, the user can type simply *d.sites* on the command line; in this case, the program will prompt the user for parameter values using the standard GRASS *parser* interface described in the manual entry for *parser*.

Parameters:

sitefile=name Name of a GRASS site_lists map in the user's current mapset search path.

color=name Sets the current color to name.
Options: red, orange, yellow, green, blue, indigo, violet, magenta, brown, gray, white, black
Default: gray

size=value Size, in pixels, in which the site icons are to be drawn.
Options: 0-1000
Default: 5

type=name The type of icon to be displayed at site locations.
Options: x, diamond, box, +
Default: x

SEE ALSO

d.display, d.icons, d.points, d.rast, d.sites, d.vect, g.region, *parser*

AUTHOR

James Westervelt, U.S. Army Construction Engineering Research Laboratory

d.sites.qual

NAME

d.sites.qual **-** Displays a subset of a sites list based on site attributes (GRASS Display Program)

GRASS VERSION

4.x, 5.x

SYNOPSIS

d.sites.qual
d.sites.qual help
d.sites.qual [-n] sitefile=name [color=name] [size=value] [type=name] [rules=name[,name,...]] [output=name]

DESCRIPTION

d.sites.qual displays sites from a sites file only if they match criteria established by rules.

Syntax of rule string is:

```
[D|C|A]#.RANGE
```

where D, C, or A specify Dimension, Category, or Attribute and RANGE is one of:

```
#-#, ge#, le#, eq#
```

examples:

```
D1.le-88.125    easting <= -88.125
D3.100-200      dimension 3 in range 100 to 200
C.eq7.2         category equal 7.2 (note no number after C)
A2.ge5000       2nd double attribute >= 5000
A1.-2.0--1.0    1st double attribute in range -2.0 to -1.0
```

If an output filename is provided, a new sites file containing the subset defined by the command will be written in the user's mapset. Header information will be copied from the original sites file and the *d.sites.qual* command and current window will be written as comments to the new file. Note that only those qualifying sites within the current window are written to the new file. The *-n* flag is useful when calling *d.sites.qual* from a script, as the number of qualifying sites may be needed to further process the file or report to the user in an interface, as in *d.siter*.

OPTIONS

This program can be run non-interactively or interactively. If the user gives the name of a GRASS site_lists map and (optionally) specifies other parameter values on the command line the program will be run non-interactively; any parameter values left unspecified are set to their default values (see below). Alternately, the user can type simply *d.sites.qual* on the command line; in this case, the program will prompt the user for parameter values using the standard GRASS *parser* interface described in the manual entry for *parser*.

Flag:

-*n*   Output number of sites displayed.

Parameters:

sitefile   Name of a site file color. Sets the current color to that stated:
Options: red, orange, yellow, green, blue, indigo, white, black, brown, magenta, aqua, gray, grey
Default: gray

size Size, in pixels, in which the icon is to be drawn.
Options: 0-1000
Default: 5

type Specify the type of the icon.
Options: x, diamond, box, +
Default: x

rules Qualifying rules for display: [D|C|A]#.RANGE where RANGE is one of #-#, ge#, le#, eq#

output Name of a site file to output drawn sites.

SEE ALSO
d.mon, d.sites, d.siter

AUTHOR
Bill Brown, UI GMS Laboratory

d.text

NAME
d.text **-** Draws text in the active display frame on the graphics monitor.
(GRASS Display Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
d.text
d.text help
d.text [size=value] [color=name] [line=value]

DESCRIPTION
d.text draws text in the active display frame on the graphics monitor. Text can be provided through standard input or redirected from a file (using the UNIX redirection mechanism).

Parameters:
size=value Height of letters, stated as a percent of the available display frame height.
Options: 0 **-** 100
Default: 5

color=name Sets display text color to name.
Options: red, orange, yellow, green, blue, indigo, violet, gray, white, black
Default: gray

line=value The screen line number on which the first line of text will be drawn.
(Line 1 is at the top of the active display frame.)
Options: 1 **-** 1000
Default: 1

In addition to the options provided on the command line, colors, text size, font type, and boldness, can be adjusted with commands in the standard input (i.e., if the user invokes *d.text* without options on the command line, and then assigns values to these options on lines within the standard input). In this case, the user also sees the standard GRASS *parser* interface described in the manual entry for *parser*.

Commands:
.*C color* (where color is one of the available colors) causes text appearing on subsequent lines to be drawn in that color.

Text size can be adjusted with the command:

.*S size* (where size is a percentage within the range 0 to 100).
Note that a size of 10 would allow 10 lines to be drawn in the active display frame, 5 would allow the drawing of 20 lines, and 50 would allow the drawing of 2 lines.

Font type can be manipulated using the command:

.*F font* (where font is one of the fonts known by the GRASS program *d.font*). Available fonts are listed in the GRASS manual entry for *d.font*. The default font type used (if unspecified by the user) is romans. Run the GRASS macro *show.fonts.sh* to see what these fonts look like.

The user can also stipulate that text be printed in bold on lines beneath the command:

.B 1 This command means bold on.

Similarly, the command:

.B 0 turns bold off of all text appearing on lines beneath it.
(Bold off is used by default, if unspecified by the user.)

EXAMPLE

The following command will print the short phrase "This is a test of d.text" in the active display frame using the color yellow, in bold, and using 4/100'ths (4%) of the active frame's vertical space per line:

```
d.text
.C yellow
.S 4
.B 1
This is a test of d.text
```

The user presses *control-d* to end input to *d.text.*

NOTES

Note that the GRASS command *d.title* creates map titles in a format suitable for input to *d.text*.

d.text needs escape sequences that can be used within lines to change colors, boldness, and perhaps size.

SEE ALSO

d.font, d.title, show.fonts.sh, parser

AUTHOR

James Westervelt, U.S. Army Construction Engineering Research Laboratory

d.title

NAME
d.title **-** Outputs a title for a raster map layer in a form suitable for display by *d.text*.
(GRASS Display Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
d.title
d.title help
d.title [-f] map=name [color=name] [size=value]

DESCRIPTION
d.title generates to standard output a string which can be used by *d.text* to draw a title for the raster map layer name in the active display frame on the graphics monitor. Output created by *d.title* can be redirected into a file, or piped directly into *d.text* to display the map title created by *d.title*. The map title created will include the map layer's name, title, MAPSET, LOCATION_NAME, geographic region boundary coordinates, and cell resolution.

The user can state program arguments on the command line, or type simply *d.title* on the command line. In the latter case, the program will prompt the user for the parameter values and flag setting using the standard GRASS *parser* interface described in the manual entry for *parser*.

OPTIONS
Flag:
-*f* Displays a fancier title.

Parameters:
map=name Name of an existing raster map layer in the user's mapset search path.

color=name Sets the current color to the name stated. Options: red, orange, green, blue, indigo, violet, black, white, gray, yellow, brown, and magenta
Default: white

size=value Sets the text size as a percentage of the active display frame's height, to value. Floating point values can be used.
Options: 0 **-** 100
Default: 15.0

EXAMPLE
For example, a user wishing to create a suitable title for the Spearfish, SD soils map layer and to display this title in the active display frame on the graphics monitor might type the following:

```
d.title map=soils color=red size=5 > title.file
```

```
d.text < title.file
```

Alternately, the user might pipe *d.title* output directly
into *d.text*:

```
d.title map=soils color=red size=5 | d.text
```

A file created by *d.title* can be displayed with *d.text.* Information contained in this file takes precedence over the color and size parameters for *d.text.*

NOTES

The text created with *d.text* will not necessarily fit within the active display frame on the graphics monitor; the user should choose a text size appropriate to this frame.

SEE ALSO

d.font, d.text, parser

AUTHORS

James Westervelt, U.S. Army Construction Engineering Research Laboratory

d.to.sites

NAME
d.to.sites **-** Allows the user to create a GRASS site_lists file by selecting points on the graphics monitor with a mouse.
(SCS GRASS Map Development Program)

GRASS VERSION
4.x

SYNOPSIS
d.to.sites
d.to.sites help
d.to.sites sites=name

DESCRIPTION
This program allows a user to create a new GRASS site_lists file by selecting locations within the active display frame. The user is prompted for the label that is placed in the site_lists file with the corresponding geographic coordinates.

OPTIONS
Parameter:

sites=name	The name to be assigned to the GRASS site_list file to be created.

EXAMPLE
For example, to create a site_lists file called SAMPLE:

d.to.sites sites=SAMPLE

The user is promted as follows:

```
Enter the label to place in the site file : SAMPLE

Buttons
Left:    Place Label Here
Middle:  Change the label
Right:   Quit
```

Note: The site_lists file must not already exist.

SEE ALSO
d.frame, d.what, d.zoom, d.sites.db.rim, s.menu

AUTHOR
M.L. Holko, USDA, SCS, NHQ-CGIS

d.vect

NAME
d.vect **-** Displays GRASS vector data in the active frame on the graphics monitor.
(GRASS Display Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
d.vect
d.vect help
d.vect map=name [color=name]

DESCRIPTION
d.vect displays the user-named binary vector file in the active display frame on the graphics monitor, in the user- named color. The vector file must already exist under a mapset listed in the user's current mapset search path.

This program can be run either non-interactively or interactively. If the user gives the name of a binary vector map to be displayed and (optionally) specifies a color for vector display, the program will run non-interactively. (If the user gives a vector map name but no vector color on the command line, the named vector map layer will be displayed in the default color white.) Alternately, the user can simply type *d.vect* on the command line; in this case, the program will prompt the user for parameter values using the standard GRASS *parser* interface described in the manual entry for *parser*.

Parameters:
map=name Vector map layer to be displayed in the active display frame. Must be binary vector file in the user's current mapset search path.

color=name Sets vector display color to name.
Options: white, red, orange, yellow, green, blue, indigo, violet, magenta, brown, gray, black
Default: white

NOTES
This program needs to be upgraded with features that would:
1) Allow different line colors, weights, and textures to be assigned to vectors with different category values.
2) Allow different fill patterns and colors to be assigned to areas.
3) Be linked with an associated program identifying these options.

SEE ALSO
d.rast, d.sites, d.vect.dlg, d.what.vect, g.mapsets, parser

AUTHOR
Michael Shapiro, U.S. Army Construction Engineering Research Laboratory

d.what.rast

NAME
d.what.rast **-** Allows the user to interactively query the category contents of multiple raster map layers at user- specified locations within the current geographic region.
(GRASS Display Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
d.what.rast
d.what.rast help
d.what.rast [-1t] [map=name[,name,...]] [fs=name]

DESCRIPTION
d.what.rast outputs the category values and labels associated with cell(s) at user-specified location(s) on user-named raster map layer(s).

The program will query the contents of raster map layer(s) named by the user on the command line. These map layers must exist in the user's current mapset search path. If the user does not name any raster map layers on the command line, *d.what.rast* will query the category contents of whatever raster map layer is already displayed in the active frame on the graphics monitor.

The program activates the mouse, and expects the user to indicate the cell location(s) to be queried by depressing a mouse button over desired location(s) within the current geographic region in the active display frame on the graphic monitor.

Flags:
-1 Identify and query just one point location. Only one mouse click is executed. This option is provided for shell scripts and programs that want to obtain only one point from the user.

-t Provide only terse output. This option is provided to simplify the parsing of output by other programs.

Parameters:
map=name Name of existing raster map layer(s).
Limit: 15 maps
Default: Query map currently displayed in the active graphics frame.

fs=name Output field separator to be used (in terse mode only).
Default: :

d.what.rast output consists of the geographic coordinates of the location pointed to, and, for each map layer, the map layer name, the category value, and category label in the named raster map layers at this location.

EXAMPLE
It is helpful, but not necessary, to first display a map to be used for reference in the active display frame before running *d.what.rast*. For example, the user might type the following series of commands and receive the output below.

d.rast map=soils
To first display the soils map in the active frame.

d.what.rast map=soils,aspect
User then moves the mouse to desired location on the displayed soils map layer, and presses the left mouse button to query the category contents of the soils and aspect maps at this geographic location. The program then outputs the below information to the user's terminal.

```
617112(E) 3732014(N)
soils in PERMANENT (44)Nunn clay loam, NdC
aspect in PERMANENT (20)15 degrees north of west
```

The first line of output gives the easting (E) and northing (N) coordinates of the geographic location at which the user clicked the mouse. The subsequent two lines give the map name and mapset, map category value (within parentheses), and map category label corresponding to this user-selected map location, for each of the maps queried by the user.

The query may be repeated as often as desired using the left mouse button. The right button on the mouse is used to quit the *d.what.rast* session.

Users can set the **-***t* flag to obtain a terse output from *d.what.rast*. This is useful when the user wishes output to be parsed by another program (like awk). If the **-***t* flag is set, users can also select the field separator used (with the *fs=name* option), or elect to use the default : field separator. In this case, the command

d.what.rast **-***t map=soils,aspect*

produces output in the form shown below. The first line of output gives the easting, northing, and the mouse button return value (see **NOTES**, below). Each subsequent line gives the map name and its mapset, the category value, and category label of the point specified on the user-named raster map layers. The default output field separator, a colon, is used since none was specified on the command line.

```
617112:3732014:1
soils@PERMANENT:44:Nunn clay loam, NdC
aspect@PERMANENT:20:15 degrees north of west
```

Users can also use this program inside of shell scripts that require as input a map category value and a mouse button depressed. Users can choose an option to run *d.what.rast* only once, and return only the map category value found and the number of the mouse button depressed.

NOTES
Mouse button return values are as follows: **0** indicates no button was pressed, **1** indicates that the left mouse button was pressed, **2** indicates the middle button was pressed, and **3** indicates that the right mouse button was pressed.

d.what.rast will always print its output to the terminal screen. *d.what.rast* output can be redirected into a file; however, if it is, the output will go both to the screen and to the file. For example:

d.what.rast map=soils,aspect > *what.out*

will both send *d.what.rast* output to the screen and capture its output in the file named what.out.

The maximum number of raster map layers that can be queried at one time is 15.

d.what.vect allows the user to interactively query the category contents of multiple vector map layers at user-specified locations.

SEE ALSO
d.rast, d.what.vect, d.where, g.region, parser

AUTHOR
Michael Shapiro, U.S. Army Construction Engineering Research Laboratory

d.what.sites

NAME
d.what.sites **-** interactively query a single site list descriptions
(GRASS Display Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
d.what.sites
d.what.sites help
d.what.sites [-1] [sitefile=name] [dcolor=color] [recolor=color]

DESCRIPTION
d.what.sites outputs the east, north coordinate values and descriptions with site(s) nearest to the user-specified location(s) on user-named site list.

The program will query the contents of site list named by the user on the command line. These site lists must exist in the user's current mapset search path.

The program activates the mouse, and expects the user to indicate the site location(s) to be queried by depressing a mouse button over desired location(s) within the current geographic region in the active display frame on the graphic monitor.

Flag:
-1 allow only one query.

Parameters:
sitefile=name Name of existing site list.

dcolor=name Sets the display color to name.
Options: red, orange, yellow, green, blue, indigo, violet, magenta, brown, gray, white, and black
Default: red

recolor=name Sets the redraw color to name.
Options: red, orange, yellow, green, blue, indigo, violet, magenta, brown, gray, white, and black
Default: yellow

size=value Size, in pixels, in which the site icons are to be drawn.
Options: 0-1000
Default: 5

type=name The type of icon to be displayed at site locations.
Options: x, diamond, box, +
Default: x

EXAMPLE
It is helpful, but not necessary, to first display a site list to be used for reference in the active display frame before running *d.what.sites*. For example, the user might type the following series of commands and receive the output below.

```
d.sites sitefile=wells
```

To first display the well site list in the active frame.

d.what.sites sitefile=wells

User then moves the mouse to desired location on the displayed well site list, and presses the left mouse button to query the category contents of the nearest well location. The program then outputs the below information to the user's terminal.

```
the closest site is: -613988.990000 east, 804343.040000 north
08376300 1968 1980 13
```

The query may be repeated as often as desired using the left mouse button. The right button on the mouse is used to quit the *d.what.sites* session. The middle button will allow to redraw the site list.

NOTES

d.what.sites will always print its output to the terminal screen. *d.what.sites* output can be redirected into a file; however, if it is, the output will go both to the screen and to the file. For example:

```
     d.what.sites sitefile=wells > what.out
```

will both send *d.what.sites* output to the screen and capture its output in the file named what.out.

SEE ALSO

d.sites, d.what.rast, d.what.vect, d.where, g.region, parser

AUTHOR

Raghavan Srinivasan, Blackland Research Center, TAES, Temple, TX 76502

d.what.vect

NAME

d.what.vect **-** Allows the user to interactively query the category contents of a (binary) vector map layer at user-selected locations within the current geographic region.
(GRASS Display Program)

GRASS VERSION

4.x, 5.x

SYNOPSIS

d.what.vect
d.what.vect help
d.what.vect [-1] map=name

DESCRIPTION

d.what.vect outputs the category value(s) associated with user-specified location(s) in a user-specified vector map layer. This program currently returns only category values for line types in the user-specified vector file.

The program will query the contents of the vector map layer named by the user on the command line. This map layer must exist in the user's current mapset search path.

The program activates the mouse, and expects the user to indicate the location(s) to be queried by depressing a mouse button over desired location(s) within the current geographic region in the active display frame on the graphic monitor.

OPTIONS

Flag:
-1 Identify and query just one point location. Only one mouse click is executed. This option is provided for shell scripts and programs that want to obtain only one point from the user.

Parameter:
map=name Name of an existing binary vector map in the user's mapset search path.

EXAMPLE

A sample *d.what.vect* session is given below. Although it is not necessary that the user first display a vector map to be queried in the active display frame, it is helpful to have a map displayed there for reference.

d.vect map=roads.24000
Displays the 1:24,000 scale roads vector map layer on the graphics monitor.

d.what.vect map=roads.24000
After typing this, the user moves the mouse to a desired location on the displayed roads map layer, and presses the left mouse button to query the category value of the roads vector map at this location. The program then outputs the category value of a line type corresponding to this user-selected map location, for the vector map queried by the user.

The query may be repeated as often as desired using the left mouse button. The right button on the mouse is used to quit the *d.what.vect* session.

Users can also use this program inside of shell scripts that require as input a map category value and a mouse button depressed. Users can set the **-***1* flag to run *d.what.vect* only once, and return only the map category value found and the number of the mouse button depressed. (Mouse button return values are as follows: **0** indicates no button was pressed, **1** indicates that the left mouse button was pressed, **2** indicates the middle button was pressed, and **3** indicates that the right mouse button was pressed.)

NOTES

Currently, *d.what.vect* only outputs category values for lines. It does not output category labels for lines, nor output category values or category labels for areas in a vector file.

d.what.vect will always print its output to the user's terminal screen. *d.what.vect* output can be redirected into a file; however, if it is, the output will go both to the screen and to the file. For example:

```
d.what.vect map=roads > what.out
```

will both send *d.what.vect* output to the screen and capture its output in the file named "what.out".

d.what.rast can be used to interactively query the map category contents of multiple raster map layers at user-selected locations.

SEE ALSO

d.rast, d.vect, d.what.rast, g.region, parser

AUTHOR

Jim Hinthorn, Central Washington University
Upgrades: Dennis Finch, National Park Service

d.where

NAME
d.where **-** Identifies the geographic coordinates associated with point locations in the active frame on the graphics monitor.
(GRASS Display Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
d.where
d.where help
d.where [-1] [spheroid=name]

DESCRIPTION
d.where is an interactive program that allows the user, using the pointing device (mouse), to identify the geographic coordinates associated with point locations within the current geographic region in the active display frame on the graphics monitor.

If the user runs *d.where* without specifying the name of a spheroid on the command line, each mouse click will output the UTM easting and northing of the point currently located beneath the mouse pointer. A mouse-button menu is presented so the user knows which mouse buttons to use. The output is always printed to the terminal screen; if the output is redirected into a file, it will be written to the file as well.

Flag:
-*l* Only one mouse click is executed. This option is provided for shell scripts and programs which want to obtain only one point from the user. The output is only written to stdout, unless redirected into a file. The geographic location and mouse button pressed are output.

Parameter:
spheroid=name Name of a spheroid (for latitude/longitude coordinate conversion).
Options: australian, bessel, clark66, clark80, everest, international, wgs72, wgs84

NOTES
This program uses the current geographic region setting and active frame. It is not necessary, although useful, to have displayed a map in the current frame before running *d.where*.

SEE ALSO
d.what.rast, d.what.vect, g.region

AUTHORS
James Westervelt, U.S. Army Construction Engineering Research Laboratory
Michael Shapiro, U.S. Army Construction Engineering Research Laboratory

d.zoom

NAME
d.zoom **-** Allows the user to change the current geographic region settings interactively, with a mouse. (GRASS Display Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
d.zoom
d.zoom help
d.zoom [-q] [action=name]

DESCRIPTION
d.zoom allows the user to interactively adjust the settings of the current geographic region using a pointing device such as a mouse. Like *g.region*, *d.zoom* re-defines the settings of the geographic region. However, *d.zoom* allows the user to change the current region settings interactively, by either outlining the new region perimeter with a mouse or "rotating" latitude/longitude data into the current region. The user should run *d.erase* after *d.zoom* is run for the new region settings to affect the graphics display.

Flag:
-*q* Run quietly, suppressing output of some program messages to standard output.

Parameter:
action=name Allows the user to change the geographic region settings by "rotating" the globe to move a different portion of the globe into the stationery boxed region outline (making this new portion of the globe the new current region), rather than by moving the boxed region outline. This option can only be used with latitudelongitude data bases (although *d.zoom* will not complain if the user attempts to set this parameter while running on a non-latitude/longitude data base).
Options: zoom, rotate

After the user types the command *d.zoom* and (optionally) sets the -*q* flag and type of zoom to be performed, a mouse-button menu will appear, directing the user to: establish the corners of the new geographic region, check its coordinates, and confirm any changes made. When the user accepts new geographic boundaries, *d.zoom* asks,

```
Accept new region?   Y / N >
```

If the user clicks the mouse over the "Y" (yes), the mouse-drawn geographic region is saved as the user's current geographic region. The user is warned that *d.erase* should be run after *d.zoom* to make new current region settings affect the graphics display.

If the user clicks the mouse over the "N" (no), the first mouse-drawn geographic region is not saved; instead, *d.zoom* asks whether or not the user wishes to

```
Try Again?   Y / N >
```

The user can then opt to draw a new region with the mouse ("Y"), or opt to exit *d.zoom* ("N") and leave current region settings unchanged.

To zoom-out to a larger area than was windowed-in on with *d.zoom* (i.e., to enlarge the current geographic region), the user may run *g.region*. The user may also alter the current geographic region by running the "region" option of the *d.display* program.

NOTES

Although it is not necessary that the user display a map in the active display frame before running *d.zoom*, it is helpful to do this for reference.

Currently, the rubber-banded boundaries of the zoom region drawn by the user are not clearly visible on the graphics monitor. However, whether or not these boundaries are clearly visible when drawn, the drawn region is still correctly zoomed-in on.

SEE ALSO

d.display, *d.erase*, *d.rast*, *g.region*

AUTHOR

Michael Shapiro, U.S. Army Construction Engineering Research Laboratory

SG3d

NAME
SG3d **-** 3 Dimensional Surface Renderer for use on Silicon Graphics workstations

GRASS VERSION
4.x, 5.x

SYNOPSIS
SG3d
SG3d help
SG3d [-w] [-z] elev=name color=name[,name,name] [vector=name] [sites=name] [3dview=name] [script=name]

DESCRIPTION
SG3d is a tool for viewing a data surface in three dimensions using GRASS on Silicon Graphics IRIS computers. At startup, one graphics window and several interface windows are visible. Other interface windows pop up when their appropriate toggle button is clicked in the Menus panel. To hide the pop up windows, click again on their toggle button in the Menus panel **-** using the Quit or Close options from the window manager's menu bar will cause the program to exit.

OPTIONS
Flags:

-*w* Enable writing to script files

-*z* Use site category as Z value (height) when displaying sites

Parameters:

elev Raster file for Elevation.

color Raster file(s) for Color (1 or 3 files). If three files are given, the first will be used for the red component, the second for the green component and the third for the blue component. When using three files, each file must contain values in the range 0 **-** 255.

vector Vector overlay file.

sites Sites overlay file.

3dview 3d viewing parameters file.

script Automatically run script file.

If the user simply types *SG3d* without specifying parameter values on the command line, the program will prompt the user to enter these.

NOTES
Environment variables SG3D_WIDTH and SG3D_HEIGHT may be set to define the width and height (in pixels) of the graphics window at startup. All SG3d windows may be resized using the window manager's normal controls at any time. Text in the control panels is not scalable, however, so the text will disappear if the panel is sized too small.

Before starting the program, check the resolution and dimensions of the current GRASS region. The current resolution will determine the best resolution available in *SG3d*. For speed, the entire raster file is loaded into memory, along with vertex color and other surface information needed for lighting calculations. Thus, large dimensions such as 1000 rows x 1000 columns may cause memory problems. *SG3d* uses 12 bytes of memory for each vertex, therefore a 1000 x 1000 image would require 12 megabytes for data alone.

The GRASS text window is used by *SG3d* to issue prompts and warnings and to input filenames. Therefore, after starting the program, place the GRASS text window where it can be seen and typed into.

Any sliders and dials used in the control panels have a FINE ADJUSTMENT: Hold down the Ctrl key while manipulating the dial or slider for fine control.

Full documentation detailing the action associated with each control is available in the tutorial.

BUGS

Sometimes very small holes will appear in the surface. If they appear as points when drawing polygons with grid over- laid, the bug is in the Graphics Library and it should be fixed in future versions of IRIX. If the holes appear along an edge of a polygon, it is apparently due to the way non- planar polygons are interpreted. In this case, drawing using the triangulate option should make the holes disappear.

AUTHORS

David Gerdes and Bill Brown, U.S. Army Construction Engineering Research Laboratory