

## GRASS PostScript Commands

[ps.icon](#)  
[ps.map](#)  
[ps.select](#)

## Other Commands

[help](#), [home](#), [database](#), [display](#), [drivers](#), [general](#), [grid3d](#), [imagery](#), [import](#), [misc](#), [models](#), [paint](#), [photo](#), [postscript](#), [raster](#), [scripts](#), [sites](#), [vector](#)

---





---

## NAME

*ps.icon* – Creates and modifies icons for use with [ps.map](#) .  
(GRASS Hardcopy PostScript Output Program)

## SYNOPSIS

`ps.icon`

## DESCRIPTION

This program allows the user to create and maintain icons which are used by [ps.map](#) to depict sites.

## FILES

Icon files created by the user are stored under \$LOCATION/ps\_icon.

## SEE ALSO

[ps.map](#)

## AUTHOR

Paul Carlson, USDA, SCS, NHQ-CGIS

*Last changed: \$Date: 2002/01/25 05:45:34 \$*




---

## NAME

*ps.map* – Hardcopy PostScript map output utility.  
(GRASS Hardcopy PostScript Output Program)

## SYNOPSIS

**ps.map**

**ps.map help**

**ps.map** [-r] [**input**=name] [**scale**=mapscale] [**copies**=n] **output**=name

## DESCRIPTION

*ps.map* produces an output file containing a PostScript program to produce hardcopy map products on your system's PostScript output device. Output can include a raster map, any number of vector overlays, site data, text labels, and other spatial data.

This program has 2 distinct modes of operation. The command-line mode requires the user to prepare a file of mapping instructions describing the various spatial and textual information to be printed prior to running *ps.map*. The interactive mode (i.e., no command-line arguments) will prompt the user for items to be mapped and does not require the user to prepare a file of instructions (some options are not used in interactive mode).

The command line flag is:

**-r**

Rotate plot 90 degrees.

The command-line parameters are:

**input**=name

File containing mapping instructions. (or enter **input**=- to enter from keyboard). These instructions are described in detail below.

**scale**=mapscale

Scale of the output map, e.g. 1:25000

Default: 1 panel

This parameter is provided as a convenience. It is identical to the *scale* mapping instruction described below.

**output**=name

Name of output the file to contain the PostScript program.

Note: the user must select a PostScript output device using [ps.select](#) before running *ps.map*.

## MAPPING INSTRUCTIONS

The mapping instructions allow the user to specify various spatial data to be plotted. These instructions are normally prepared in a regular text file using a system editor. Some instructions are single line instructions while others are multiple line. Multiple line instructions consist of the main instruction followed by a subsection of one or more additional instructions.

Instructions that may be included in the subsection under several different main instructions are:

*where* \ *x* \ *y*

The top left corner of the bounding box of the item to be plotted is located *x* inches from the left edge of the paper and *y* inches from the top edge of the paper. If *x* is less than or equal to zero, the default horizontal location is used. If *y* is less than or equal to zero, the default vertical location is used.

*font* \ *font* \ *name*

The name of the PostScript font. The default is *Helvetica*.

*fontsize* \ *font* \ *size*

The size of the PostScript font in 1/72 inch. The default is 10.

## colortable

Prints the color table for the raster map layer anywhere on the page.

```
USAGE:          colortable [y|n]
where x y
width table width
height table height (FP legend only)
cols table columns
font font name
fontsize font size
color text color
nodata [y|n]
end
```

The color table will display the colors for each raster map layer category value and the category label. To get a color table, you must have previously requested a raster map layer. The default location for the colortable is immediately below any other map legend information, starting at the left margin. The default text color is black. Omitting the **colortable** instruction would result in no color table. For floating point legends **width** is width of color band only. **height** is used only for floating point legend.

**Note:** Be careful about asking for color tables for raster map layers which have many categories, such as elevation. This could result in the printing of an extremely long color table!!!!

Another issue is that the color table only includes categories which have a label. If there are only a few categories, you can use `r.support` to manually add labels. If there are too many categories to do this, you could write a script to add dummy labels to the cats file (`<gisdbase>/<location>/<mapset>/cats/<mapname>`).

This example would print a color table immediately below any other map legend information, starting at the left margin.

```
EXAMPLE:
colortable y
```

## comments

Prints comments anywhere on the page.

```
USAGE:      comments commentfile
where x y
font font name
fontsize font size
color text color
end
```

The default location is immediately below the last item item printed, starting at the left margin. The default text color is black.

This example prints in blue whatever is in the file *veg.comments* starting at 1.5 inches from the left edge of the page and 7.25 inches from the top of the page, using a 15/72 inch Helvetica Bold font.

```
EXAMPLE:
raster vegetation
comments veg.comments
where 1.5 7.25
font Helvetica Bold
fontsize 15
color blue
end
```

Presumably, the file *veg.comments* contain comments pertaining to the raster map layer *vegetation*, such as "This map was created by classifying a LANDSAT TM image".

## copies

Specifies the number of copies to be printed.

```
USAGE:      copies n
```

Each page will be printed n times.

## eps

Places EPS (Encapsulated PostScript) pictures on the output map.

```
USAGE:      eps east north
eps x% y%
epsfile EPS file
scale #
rotate #
masked [y|n]
end
```

The EPS picture location is entered in the main instruction line by giving either the map coordinates or by using percentages of the geographic region. The user must specify full EPS file path **epsfile**. The user may also specify the **scale** of the icon (default is 1.0), the **rotate** i.e. rotation in degrees (default is 0) and whether the point is to be **masked** by the current mask. (See manual entry for [r.mask](#) for more information on the

mask.)

This example would place a EPS file `./epsf/logo.eps` at the point (E456000 N7890000). This picture would be rotated 20 degrees clockwise, 3 times bigger than in original file and would not be masked by the current mask.

```
EXAMPLE:
eps 456000 7890000
epsfile ./epsf/logo.eps
scale 3
rotate 20
masked n
end
```

Of course, multiple EPS pictures may be drawn with multiple *eps* instructions.

## greyrast

Selects a raster map layer for output in shades of grey.

```
USAGE:      greyrast mapname | list
```

For each *ps.map* run, only one raster map layer can be requested (using either the *greyrast* or the *raster* instruction).

## grid

Overlays a coordinate grid onto the output map.

```
USAGE:      grid spacing
color color
numbers # [color]
font font name
fontsize font size
end
```

The **spacing** of the grid is given (in the geographic coordinate system units) on the main instruction line. The subsection instructions allow the user to specify the **color** of the grid lines, whether coordinate **numbers** should appear on the grid lines, and if they should appear every grid line (1), every other grid line (2), etc., and what color the numbers should be. The defaults are black grid lines, unnumbered.

This example would overlay a green grid with a spacing of 10000 meters (for a metered database, like UTM) onto the output map. Alternate grid lines would be numbered with red numbers.

```
EXAMPLE:
grid 10000
color green
numbers 2 red
end
```

## group

Selects an RGB imagery group for output.

```
USAGE:      group groupname
```

This is similar to *raster*, except that it uses an imagery group instead of a raster map layer. The group must contain three raster map layers, comprising the red, green and blue bands of the image.

## header

Prints the map header above the map.

```
USAGE:      header
file header file
font font name
fontsize font size
color text color
end
```

If the *header* instruction or the *file* sub-instruction is absent, the header will consist of the map TITLE and location, each centered on the page above the map. The default text color is black.

This example prints (in red) whatever is in the file *soils.hdr* above the map, using a 20/72 inch Courier font.

```
EXAMPLE:
header
file soils.hdr
font Courier
fontsize 20
color red
end
```

## labels

Selects a labels file for output (see manual entry for [p.labels](#)).

```
USAGE:      labels labelfile|list
font font name
end
```

NOTE: ps.map can read new option 'ROTATE:' from labels file, which specifies counter clockwise rotation in degrees.

This example would paint labels from the labels file called *town.names*. Presumably, these labels would indicate the names of towns on the map.

```
EXAMPLE:
labels town.names
```

## line

Draws lines on the output map.

```
USAGE:      line east north east north
line x% y% x% y%
color color
width #
masked [y|n]
end
```

The beginning and ending points of the line are entered on the main instruction. These points can be defined either by map coordinates or by using percentages of the geographic region. The user may also specify line **color**, **width** in pixels (accepts decimal points [floating points] as well as integers), and if the line is to be **masked** by the current mask. (See manual entry for [r.mask](#) for more information on the mask.)

This example would draw a yellow line from the point x=10% y=80% to the point x=30% y=70%. This line would be 2 pixels wide and would appear even if there is a mask.

```
EXAMPLE:
line 10% 80% 30% 70%
color yellow
width 2
masked n
end
```

Of course, multiple lines may be drawn with multiple *line* instructions.

## mapinfo

Prints the portion of the map legend containing the scale, grid and region information, on or below the map.

```
USAGE:      mapinfo
where x y
font font name
fontsize font size
color text color
end
```

The default location is immediately below the map, starting at the left edge of the map. The default text color is black.

This example prints (in brown) the scale, grid and region information immediately below the map and starting 1.5 inches from the left edge of the page, using a 12/72 inch Courier font.

```
EXAMPLE:
mapinfo
where 1.5 0
font Courier
fontsize 12
color brown
end
```



## maploc

Positions the map on the page.

```
USAGE:      maploc x y [width height]
```

The upper left corner of the map will be positioned *x* inches from the left edge of the page and *y* inches from the top of the page. If *width* and *height* (in inches) are present, the map will be rescaled, if necessary, to fit.

This example positions the upper left corner of the map 2.0 inches from the left edge and 3.5 inches from the top edge of the map.

```
EXAMPLE:
maploc 2.0 3.5
```

## outline

Outlines the areas of a raster map layer with a specified color.

```
USAGE:      outline
color color
width width of line in pixels
end
```

Distinct areas of the raster map will be separated from each other visually by drawing a border (or outline) in the specified **color** (default: black). For **width** the program accepts decimal points [floating points] as well as integers. Note: it is important the user enter the instruction **end** even if a color is not chosen. (It is hoped that in the future the outline of a different raster map layer other than the one currently being painted may be placed on the map.)

This example would outline the category areas of the *soils* raster map layer in grey.

```
EXAMPLE:
raster soils
outline
color grey
width 2
end
```

## point

Places additional points or icons on the output map.

```
USAGE:      point east north
point x% y%
color color
icon iconfile|list
size #
masked [y|n]
end
```

The point location is entered in the main instruction line by giving either the map coordinates or by using percentages of the geographic region. The user may also specify the point **color**, the **icon** file to be used to

## GRASS PostScript Commands

represent the point location (see the the manual entry for [ps.icon s](#)), the **size** of the icon as a multiple of the size of the pattern in the icon file (default is 1.0, approximately equivalent to a 10 point character), and whether the point is to be **masked** by the current mask. (See manual entry for [r.mask](#) for more information on the mask.)

This example would place a purple diamond (from icon file *diamond*) at the point (E456000 N7890000). This diamond would be the approximately the size of a 15 point character and would not be masked by the current mask.

```
EXAMPLE:  
point 456000 7890000  
color purple  
icon diamond  
size 1.5  
masked n  
end
```

Of course, multiple points may be drawn with multiple *point* instructions.

## psfile

Copies a file containing PostScript commands into the output file.

**Note:** *ps.map* will not search for this file. The user must be in the correct directory or specify the full path on the **psfile** instruction. (Note to /bin/csh users: ~ won't work with this instruction).

```
USAGE:      psfile filename
```

This example copies the file "logo.ps" into the output file.

```
EXAMPLE:  
psfile logo.ps
```

## raster

Selects a raster map layer for output.

```
USAGE:      raster mapname | list
```

For each *ps.map* run, only one raster map layer (or set of layers or imagery group; see below) can be requested. If no raster map layer is requested, a completely white map will be produced. It can be useful to select no raster map layer in order to provide a white background for vector images.

Note that an imagery group selected with the *group* option, or a set of three raster layers selected with the *rgb* option, count as a raster map layer for the purposes of the preceding paragraph.

This example would paint a map of the raster map layer *soils*.

```
EXAMPLE:  
raster soils
```

## read

Provides *ps.map* with a previously prepared input stream.

USAGE:           **read** previously prepared UNIX file

Mapping instructions can be placed into a file and read into *ps.map*.

**Note:** *ps.map* will not search for this file. The user must be in the correct directory or specify the full path on the **read** instruction. (Note to /bin/csh users: ~ won't work with this instruction).

This example reads the UNIX file *pmap.roads* into *ps.map*. This file may contain all the *ps.map* instructions for placing the vector map layer *roads* onto the output map.

EXAMPLE:  
**read** pmap.roads

The user may have created this file because this vector map layer is particularly useful for many *ps.map* outputs. By using the **read** option, the user need not enter all the input for the **vector** instruction, but simply **read** the previously prepared file with the correct instructions.

## rectangle

Draws rectangle on the output map.

USAGE:           **rectangle** east north east north  
**rectangle** x% y% x% y%  
**color** color  
**fcolor** fill color  
**width** #  
**masked** [y|n]  
**end**

The two corners of the rectangle are entered on the main instruction. These points can be defined either by map coordinates or by using percentages of the geographic region. The user may also specify line **color**, fill color **fcolor**, **width** in pixels (accepts decimal points [floating points] as well as integers), and if the rectangle is to be **masked** by the current mask. (See manual entry for [r.mask](#) for more information on the mask.)

This example would draw a yellow rectangle filled by green from the point x=10% y=80% to the point x=30% y=70%. This line would be 2 pixels wide and would appear even if there is a mask.

EXAMPLE:  
**rectangle** 10% 80% 30% 70%  
**color** yellow  
**fcolor** green  
**width** 2  
**masked** n  
**end**

Of course, multiple rectangles may be drawn with multiple *rectangle* instructions.

## region

Places the outline of a smaller geographic region on the output.

```
USAGE:      region regionfile|list
color color
width #
end
```

Geographic region settings are created and saved using [g.region](#). The *ps.map region* option can be used to show an outline of a smaller region which was printed on a separate run of *ps.map* on other user-created maps.

The user can specify the **color** and the **width** in pixel units (accepts decimal points [floating points] as well as integers) of the outline. The default is a black border of one pixel width.

This example would place a white outline, 2 pixels wide, of the geographic region called *fire.zones* onto the output map. This geographic region would have been created and saved using [g.region](#).

```
EXAMPLE:
region fire.zones
color white
width 2
end
```

## rgb

Selects three raster map layers for output as an RGB color image.

```
USAGE:      rgb red green blue
```

This is similar to *raster*, except that it uses three raster map layers instead of a single layer. The three layers are composed to form a color image, similar to *d.rgb*.

For each layer, only one of the components of the layer's color table is used: the red component for the red layer, and so on. This will give the desired result if all of the layers have a grey-scale color table, or if each layer's color table uses the hue appropriate to the layer.

## scale

Selects a scale for the output map.

```
USAGE:      scale scale
```

The scale can be selected either as:

a relative ratio, e.g. 1:25000;

an absolute width of the printed map, e.g. 10 inches;

the number of printed paper panels, e.g. 3 panels *.I* (at the present time, only 1 panel is supported);

the number of miles per inch, e.g. 1 inch equals 4 miles.

This example would set the scale of the map to 1 unit = 25000 units.

EXAMPLE:  
**scale** 1:25000

## setcolor

Overrides the color assigned to one or more categories of the raster map layer.

USAGE:           **setcolor** cat(s) color

This example would set the color for categories 2,5 and 8 of the raster map layer *watersheds* to white and category 10 to green. (**NOTE**: no spaces are inserted between the category values.)

EXAMPLE:  
**raster** watersheds  
**setcolor** 2,5,8 white  
**setcolor** 10 green

Of course, *setcolor* can be requested more than once to override the default color for additional categories. More than one category can be changed for each request by listing all the category values separated by commas (but with no spaces).

## sites

Selects sites data to be placed on the output map (see manual entry for [s.menu](#)).

USAGE:           **sites** sitemap|list  
**color** color  
**icon** iconfile|list  
**eps** epsfile  
**size** #  
**size\_att** #  
**rotate** #  
**desc** [y|n]  
**font** font name  
**end**

The user may specify the the **color** of the sites (see section on NAMED COLORS below); the **icon** to be used to represent the presence of a site (see the manual entry for [p.icons](#)); the **eps** Encapsulate Postscript file to be used to represent the presence of a site; If \$ is used in EPS file path it is replaced by category number. the **size** of the icon (number of times larger than the size it is in the icon file); the **size\_att** number of attribute used for size (if used size is equal to size \* attribute\_value); the **rotate** in degrees for clockwise rotation for EPS files; and whether or not the **description** associated with each site is also to be printed. If the **description** is to be printed, the **font** name may be specified. The size of the font is proportional to the icon **size**. This example would paint a sites map with blue windmills (from an icon file created by the user using the [p.icons](#) GRASS command) placed at all windmill locations (from a sites list). These windmills would be two times larger than the size of the icon in the icon file and have descriptions from the sites list file printed beside them.

EXAMPLE:  
**sites** windmills  
**color** blue  
**icon** windmill  
**size** 2  
**desc** y  
**end**

## text

Places text on the map.

```

USAGE:      text east north text
text x% y% text
font fontname
color color|none
width #
hcolor color|none
hwidth #
background color|none
border color|none
size #
ref reference point
xoffset #
yoffset #
opaque [y|n]
end

```

The user specifies where the text will be placed by providing map coordinates or percentages of the geographic region map. The text follows these coordinates on the same instruction line. More than one line of text can be specified by notating the end of a line with `\n` (e.g. USA\\nCERL).

The user can then specify various text features:

**font:** cyrilc gothgbt gothgrt gothitt greekc greekcs greekp greeks italicc italiccs italicc romanc romancs romand romans romant scriptc scripts (The default font is romans);

**color** (see NAMED COLORS);

**width** of the lines used to draw the text to make thicker letters (accepts decimal points [floating points] as well as integers);

**size** as the vertical height of the letters in meters on the ground (text size will grow or shrink depending on the scale at which the map is painted);

the highlight color (**hcolor**) and the width of the highlight color (**hwidth**);

the text-enclosing-box **background** color; the text box **border** color;

**ref.** This reference point specifies the text handle – what part of the text should be placed on the location specified by the map coordinates. Reference points can refer to: [lower|upper|center] [left|right|center] of the text to be printed;

**yoffset**, which provides finer placement of text by shifting the text a vertical distance in pixels from the specified north. The vertical offset will shift the location to the south if positive, north if negative;

**xoffset**, which shifts the text a horizontal distance in pixels from the specified east. The horizontal offset will shift the location east if positive, west if negative;

whether or not the text should be **opaque** to vectors. Entering **no** to the opaque option will allow the user to see any vectors which go through the text's background box. Otherwise, they will end at the box's edge.

## GRASS PostScript Commands

This example would place the text *SPEARFISH LAND COVER* at the coordinates E650000 N7365000. The text would be a total of 3 pixels wide (2 pixels of red text and 1 pixel black highlight), have a white background enclosed in a red box, and be 500 meters in size. The lower right corner of the text would be centered over the coordinates provided. All vectors on the map would stop at the border of this text.

```
EXAMPLE:
text 650000 7365000 SPEARFISH LAND COVER
font romand
color red
width 2
hcolor black
hwidth 1
background white
border red
size 500
ref lower left
opaque y
end
```

## vector

Selects a vector map layer for output.

```
USAGE:          vector vectormap|list
color color
acolor r g b
width #
cwidth #
hcolor color
hwidth #
offset #
coffset #
ref left|right
masked [y|n]
style 0-9
line_cat #
label label
lpos 0|1-20
pat name
pwidth #
scale #0
end
```

The user can specify:

**color** – color of the vector lines or area boundaries;

**acolor** – the area color. If **acolor** is used areas are plotted instead of lines (accepts 3 integers from 0 to 255 for red, green and blue separated by spaces). This color is also used by pattern until overwritten in pattern file;

**width** – width of the vectors lines or area boundaries in pixels (accepts decimal points [floating points] as well as integers);

**cwidth** – width of the vectors lines. If **cwidth** is used then width of line is equal to **cwidth** \* category value and **width** is used in legend;

## GRASS PostScript Commands

**hcolor** – the highlight color for the vector lines;

**hwidth** – the width of the highlight color in pixels;

**offset** (experimental) – offset for the vectors lines in pixels for plotting parallel lines in distance equal to offset (accepts positive or negative decimal points);

**coffset** (experimental) – offset for the vectors lines. If coffset is used then offset of line is equal to coffset \* category value and offset is used in legend;

**ref** (experimental) – line justification.

**masked** – whether or not the raster map layer is to be masked by the current mask (see manual entry [r.mask](#) for more information on the mask);

**style** – the line style allows the vectors to be dashed in different patterns. This is done by typing a series of numbers (0's and 1's) in a desired sequence or pattern. Blanks and non-digit characters are recognized as 0's. Using 0 would allow the colors of the raster map layer (or the background color if no raster map layer was selected) to show through;

**line\_cat** – which category should be plotted (default is all);

**label** – for description in vlegend. Default is: map(mapset);

**lpos** – position vector is plotted in legend. If lpos is 0 then this vector is omitted in legend. If more vectors used the same lpos then their symbols in legend are merged and label for first vector is used.

**pat** – full path to pattern file. Pattern file contains header and simple PS commands. It is similar to EPS but more limited, that means that each pattern file is EPS file but EPS files are not usually useful as pattern files because contain restricted commands. Color and width of patterns is set by **acolor** and **pwidth** until it is overwritten in pattern file. Currently the only way to create pattern file is text editor. Example of pattern file:

```
%!PS-Adobe-2.0 EPSF-1.2
%%BoundingBox: 0 0 10 10
newpath
5 0 moveto
5 10 lineto
stroke
```

**scale** – pattern scale

**pwidth** – pattern line width, width is used by pattern until the width is overwritten in pattern file.

This example would paint a map of the vector map layer named *streams*. These streams would be a total of 3 pixels wide (the inner two pixels blue and the outer highlight pixel white). The map would not show streams outside of the current mask. Only streams of category 2 will be plotted. In legend appears label 'Streams – category 2'.

```
EXAMPLE:
vector streams
color blue
width 2
hcolor white
```



```

hwidth 1
masked y
line_cat 2
label Streams - category 2
end

```

## verbose

Changes the amount of talking *ps.map* will do.

USAGE: **verbose** [0|1|2]

A higher value implies more chatter. The default is 2. This example sets the amount of chatter to a minimum.

EXAMPLE:  
**verbose** 0

## vlegend

Prints the portion of the map legend containing the vector information, on or below the map.

```

USAGE:          vlegend
where x y
font font name
fontsize font size
end

```

The default location is immediately below the legend containing the scale, grid and region information, starting at the left edge of the map. If the *where* instruction is present and *y* is less than or equal to zero, the vector legend will be positioned immediately below the map, starting *x* inches from the left edge of the page.

This example prints the vector legend immediately below the map and starting 4.5 inches from the left edge of the page, using a 12/72 inch Helvetica font.

EXAMPLE:  
**vlegend**  
**where** 4.5 0  
**font** Courier  
**fontsize** 12  
**end**

## end

Terminates input and begin painting the map.

USAGE: **end**

## NAMED COLORS

The following are the colors that are accepted by *ps.map*:

aqua

```

black
blue
brown
cyan
gray
green
grey
indigo
magenta
orange
purple
red
violet
white
yellow

```

## EXAMPLE ps.map INPUT FILE

The following is an example of a *ps.map* script file. The file has been named *spear.soils*. For the purposes of illustration, the file is in two columns. This script file can be entered at the command line:

```
ps.map input=spear.soils output=soils.ps
```

```

raster soils
vector roads
  width 2
  style 0011
  color 1 black
  color 2 red
  masked n
end
labels town.names
region subregion
  color white
  width 2
end
grid 10000
  color green
  numbers 2 red
end
outline
  color black
end
colortable y
comments soil.cmt
  font Helvetica
end
scale 1:25000
setcolor 6,8,9 white
setcolor 10 green
vlegend
  where 4.5 0
end
text 608000.00 3476004.73 SPEARFISH SOILS MAP
  color red
  width 2
  hcolor black
  hwidth 1
  background white

```

```
border red
size 500
ref lower left
opaque y
end
line 606969.73 3423092.91 616969.73 3423092.91
color yellow
width 2
opaque yes
end
point 40% 60%
color purple
icon diamond
size 2
masked n
end
end
```

## INTERACTIVE MODE

If the user simply enters *ps.map* without arguments, then a simple prompting session occurs. Some, but not all of the non-interactive requests are available at this level.

## NOTES

The user can specify negative values for position of EPS-files in *ps.map* to move them outside the current region (to position a barscale or other legend entries).

## SEE ALSO

[\*ps.icon\*](#)  
[\*ps.select\*](#)

## AUTHOR

Paul Carlson, USDA, SCS, NHQ-CGIS  
Modifications: Radim Blazek, Glynn Clements

*Last changed: \$Date: 2003/02/13 20:17:32 \$*



## NAME

*ps.select* – Selects a PostScript device for GRASS hardcopy output.  
(*GRASS Hardcopy PostScript Output Program*)

## SYNOPSIS

```
ps.select
ps.select help
ps.select [ -lpq] [painter=name]
```

## DESCRIPTION

**ps.select** allows the user to select the PostScript device for GRASS hardcopy output. If the user has not selected a device *before* running [ps.map](#), the default device configuration (shown under **NOTES** below, will be used.

## OPTIONS

### Flags:

- l* List all available PostScript painters.
- p* Print name of currently selected PostScript painter.
- q* Quietly select PostScript painter.

### Parameters:

*painter*=name  
Name of PostScript painter to select.  
Options: (this will be a list of available hardcopy output PostScript devices)

## INTERACTIVE MODE

If the user runs *ps.select* without specifying program arguments on the command line, the program will prompt the user for the name of a hardcopy output PostScript device to select.

## NOTES

Each PostScript device must have a device description file in the `$GISBASE/etc/paint/ps\*.devices` directory. The name of the device description file is used as the name of the PostScript painter. A device description file contains the following information (default values are shown):

```
level: 2
page width: 8.5
page height: 11.0
top margin: 0.5
bottom margin: 0.5
left margin: 0.25
right margin: 0.25
resolution: 75
```

where *level* is the PostScript level supported by the device (1 or 2); *page width* and *page height* are the paper dimensions (in inches); *top margin*, *bottom margin*, *left margin* and *right margin* are the unprintable edges of the page (in inches); and *resolution* is the effective resolution (in pixels per inch).

## SEE ALSO

[ps.map](#)

## AUTHOR

Paul Carlson, USDA, SCS, NHQ-CGIS

*Last changed: \$Date: 2002/01/25 05:45:34 \$*