

NAME

`gri` – scientific graphics language

SYNOPSIS

`gri` [**OPTIONS**] [*CommandFile* [*optional_arguments*]]

DESCRIPTION

Gri is a programming language for scientific graphics. It can make x-y graphs, contour-graphs, and image graphs. In addition, Gri has a full suite of low-level graphical elements and sufficient programming capabilities (loops, subroutines, etc) to permit complex customization.

Gri is not a point-click application. In some ways it is analogous to TeX. Extensive power rewards tolerance of a modest learning curve.

For more information, please consult online *info* and *html* manuals.

The *info* manual is normally accessed by typing

info gri

The *html* manual is located at `/usr/doc/gri-N.N.N/html/index.html`, where N.N.N is the version number.

The *html* FAQ is located at `/usr/doc/gri-N.N.N/html/FAQ.html`

GRI_MERGE AND GRI_UNPAGE COMMANDS

Two Perl scripts are provided with Gri to manipulate the PostScript output.

gri_merge is used to merge multiple Gri output files into a single PostScript file. Type **gri_merge -h** for usage information.

gri_unpage is used to split a multi-page Gri output file (in which the **new page** command was used) into separate PostScript files, one for each page. Type **gri_unpage -h**

EMACS SUPPORT

An *emacs* mode is provided with Gri.

The mode is installed automatically in Debian by the elisp file:

`/etc/emacs/site-start.d/50gri.el`

The emacs mode itself is *gri-mode.el* and is installed on Debian as `/usr/share/emacs/site-lisp/gri-mode.el`

Byte-compiled versions of this file are produced for every flavour of Emacs that is installed, and are located in places like `/usr/share/emacs/20.2/site-lisp/gri-mode.elc`

AUTHOR

Gri (c) 1999-2002 Dan Kelley <Dan.Kelley@Dal.CA>

This manual page by Peter S Galbraith <psg@debian.org>.

NAME

`gri_merge` – merge multiple Gri output files into a single PostScript file.

USAGE (style 1):

```
gri_merge [OPTIONS] CxR a.ps b.ps ... > merged_file.ps
```

Merges the files onto one page, in 'C' columns and 'R' rows. The CxR files are given in the order of words on a page. The page is presumed to be 8.5x11in in size, as are all the input files, and the input files are sized to fit, and kept in natural scale.

USAGE (style 2):

```
gri_merge [OPTIONS] xcm ycm enlarge a.ps [b.ps ...] > merged_file.ps
```

Where 'enlarge' is a scale factor applied after offsetting 'xcm' to the right and 'ycm' upward.

EXAMPLE (style 2):

The following

```
gri_merge 2 12 .5 a.ps \
12 12 .5 b.ps \
2 2 .5 c.ps \
12 2 .5 d.ps > all.ps
```

produces 4 panels from gri plots done using margins and sizes as specified in the following lines in a gri commandfile

```
set x margin 2
```

```
set x size 15
```

```
set y margin 2
```

```
set y size 15
```

The OPTIONS, available if your 'perl' has 'getopts' library, are:

-u graylevel — set graylevel for underlay beneath panels, by default 0.75.

Values range from 0 (black) to 1 (white), although a value of precisely 1 means do NOT draw underlay.

-b graylevel — Set value for background under individual panels, again 0 for black to 1 for white, with 1 meaning no drawing.

-h — Print this help message and quit.

SEE ALSO

gri(1), gri_unpage(1)

NAME

`gri_unpage` – split a multi-page Gri output file into separate files.

DESCRIPTION

`gri_unpage` is used to split a multi-page Gri output file (in which the **new page** command was used) into separate PostScript files, one for each page.

USAGE

```
# gri_unpage name.ps
```

Creates files `name-1.ps`, `name-2.ps`, etc, one for each page of `name.ps`.

OPTIONS:

The OPTIONS, available if your 'perl' has 'getopts' library, are:

-l — sets the bounding box to letter page size.

By default, the bounding box size is extracted from the original file, and is used for all newly created files.

BUGS:

1. Bounding box is always the same size (grabbed from the original).
2. Assumes that all Gri fonts are used even if they aren't.

SEE ALSO

`gri(1)`, **`gri_merge(1)`**