

NAME

peps – manual page for peps v.2.0

SYNOPSIS

peps [*option ...*] [*file*]

xpeps [*option ...*] [*file*]

DESCRIPTION

Peps preprocesses an encapsulated Postscript (**EPS**) file and passes it on to *Ghostscript* for a conversion into a bitmap. In version 1.0 the default output was a *pnm* bitmap, or optionally a *PNG* bitmap. As of version 2.0, **peps** default output is a *PNG* bitmap with 16 million colors, grayscale with the **-m** switch, while a *pnm* bitmap is produced when the **-n** switch is specified. Additionally, any other bitmap format supported by *Ghostscript* can be requested using the **-f** switch. **Xpeps** default output is a color image produced by the *Ghostscript x11* device, a grayscale image produced by the *x11gray4* device with the **-m** switch.

While Postscript does not use the **#** as a comment delimiter, **peps** will not pass the *first* line of its input to *Ghostscript*, if that line starts with a **#**. This is to allow the Unix shell to treat encapsulated Postscript files as executable scripts.

Peps can be used on line to produce images on the fly using the standard **CGI** interface. **Xpeps** can be used to debug online **peps** scripts before posting them on line.

OPTIONS

The following command line options control **peps** and **xpeps** behavior. If several contradicting options are used, the later ones override the earlier ones. Both **peps** and **xpeps** accept the same switches. However, each will ignore those switches that do not apply to their operation: **Peps** will ignore **-e** and **-E**, while **xpeps** ignores **-c**, **-C**, **-f**, **-M**, **-n**, **-u**, **-z** and **-Z**. Nevertheless, even if a switch is ignored, as long as it is used, it must be used correctly. For example, **xpeps** will complain if **-f** is not followed by a syntactically correct name of a *Ghostscript* device.

In the description of the switches, unless **xpeps** is mentioned explicitly, any time **peps** is discussed, the discussion also applies to **xpeps**.

-a <*angle*>

Declare the *angle of rotation* of the image in degrees, about the center of the image. By default the angle is **0**, that is no rotation. A positive angle rotates the image *counter-clockwise*, a negative one *clockwise*.

Rotation almost always results in a part of the image being cut off. This can be prevented by using the **-x**, **-X**, **-y**, and **-Y** switches, or by changing the bounding box with the **-b** switch.

-b <*left*> <*bottom*> <*right*> <*top*>

Override the bounding box, which, by default, is read from the EPS file header. When this switch is used, **peps** completely ignores the EPS file header. That means the input does not even need to have an EPS header.

-c Print HTTP content type, i.e. the MIME type. **Xpeps** always ignores this option. **Peps** ignores it if you select an output file instead of the default *stdout*. The content type depends on the type of bitmap produced. **Peps** will print the HTTP content type followed by **two** new lines into *stdout* and only then will it instruct Ghostscript to produce the bitmap and to send it to *stdout*. This lets you use **peps** as a CGI program on web sites to produce on the fly images.

Additionally, if gzip compression is turned on, whether by default or with the **-z** switch, **peps** will also print the correct HTTP content encoding to *stdout*.

-C Print HTTP content type and exit. Like with the **-c** option, **peps** will print the MIME type to *stdout*. Unlike that, however, it will follow the MIME type by only **one** new line, and then it will exit without processing any input. This is useful for CGI programming when you want full control of the HTTP header. If **xpeps** sees this switch, it just exits.

-d Show defaults. Since system administrators can compile **peps** and **xpeps** with different defaults, you can use this switch to see what defaults are used on your system.

-e [*time*]

By default, **xpeps** waits for you to press enter before removing the image from the display. With this option, **xpeps** will run in the background and will wait for *time* seconds before removing the image. If you do not specify the *time*, **xpeps** will use the default *time* to wait. This is a very long time (68 years on 32-bit machines), though your system administrator may have compiled **xpeps** with a shorter default. You can find out the default *time* by running

xpeps -d

This option has no effect on **peps**.

-E [*time*]

Same as **-e**, but **xpeps** will also print its *pid* on *stdout*. This will allow you to *kill xpeps* when you no longer need to see the image. Please note that just closing the window showing the image will not automatically *kill xpeps*, you need to *kill* it explicitly, or wait till *time* seconds expire (which can be never if you do not specify *time* explicitly).

-f <*format*>

Output in a different bitmap format, where *format* is a Ghostscript bitmap/pixmap output device. This option has no effect on **xpeps**.

-g 1 | 2 | 4

Anti-aliasing level of graphics. **1** is *no* anti-aliasing, **2** is *some* anti-aliasing, **4** is *full* anti-aliasing (default is **4**). See also **-l** and **-t**.

-h <*resolution*>

Horizontal resolution in ppi (default **100**). Overrides **-w** and **-W**, as well as the horizontal resolution set by **-r**.

-h Stop processing the command line, show help and exit. Unlike the option above, it takes no parameters, so **peps** can distinguish between you asking for help and you setting the horizontal resolution.

-i Use *stdin* for input. This is the default in version 2.0. Before that, **peps** did not accept input from *stdin* at all. There was a reason for that: While EPS files must have an EPS header before they start the actual Postscript code, the EPS specification allows the creators of EPS files to defer the *%%BoundingBox* comment to the trailer. That presents no problem with files which **peps** can process and rewind. But it does present a problem when reading the input from *stdin* which cannot be rewound.

As of version 2.0, **peps** accepts its input from *stdin* but in that case it requires that either the *%%BoundingBox* comment be in the header, or that you use the **-b** switch and set the bounding box explicitly.

-l 1 | 2 | 4

Anti-aliasing level of both, graphics and text. **1** is *no* anti-aliasing, **2** is *some* anti-aliasing, **4** is *full* anti-aliasing (default is **4**). It is a shortcut for **-g** and **-t** when you want to set the same anti-aliasing level for both.

-m **Peps** will output in monochrome (grayscale) *PNG* format. **Xpeps** will use the *x11gray4 Ghostscript* device (unless compiled with a different default).

-M <*file*>

If the **-c** or the **-C** switch is used, **peps** will look into *file* in search of the proper MIME type for the output format before it looks anywhere else. See the **MIME** section below for more details. This option has no effect on **xpeps**.

- n** Output in *pnm* format. This was the default in version 1.0. This option has no effect on **xpeps**.
- o <file>**
Output to a file named *file*. By default **peps** sends its output to *stdout*. You can also request *stdout* explicitly by entering an empty file name, i.e. **peps -o ""** or **peps -o "**.
- p** Output in color *PNG* format with **peps**, or to the *x11* device with **xpeps**. As of version 2.0, this is the default on most systems though the system administrator has the option of using the *pnm* format with **peps** as the default, so it is a good idea to use the **-p** switch explicitly.
- q <height>**
The output will be *height* pixels high. This overrides the vertical resolution set by the **-v** or **-r** switches and discards any overrides of vertical resolution made by the **-Q** or **-W** switches. By default the height of the bitmap is determined mathematically from the vertical resolution and the height of the bounding box. With this switch the vertical resolution is calculated from the *height* in pixels and the height of the bounding box. This switch does not affect the width of the bitmap or the horizontal resolution set by the **-h**, **-r**, **-w** or **-W** switches, but it does replace any override of the horizontal resolution made by the **-Q** switch. In other words, **-Q 500 -q 100** has the same effect as **-Q 100**.
- Q <height>**
Does everything the **-q** switch does, but also affects the width of the bitmap by setting the horizontal resolution the same as the new vertical resolution. As a result, the bitmap will be *height* pixels high while keeping its height/width ratio as determined by the bounding box. This switch overrides the **-h**, **-q**, **-r** and **-v** switches, and discards any overrides made by the **-w** and **-W** switches.
- r <resolution>**
Vertical and horizontal resolution in ppi (default **100**). Overrides **-h**, **-q**, **-Q**, **-v**, **-w** and **-W**.
- s** Turns off Ghostscript safety. By default **peps** passes the **-dSAFER** switch to Ghostscript which disables unsafe Postscript commands. With the **-s** option, **peps** will **not** use that Ghostscript switch. In most cases the use of this option should be avoided.
- t 1 | 2 | 4**
Antialiasing level of text. **1** is *no* anti-aliasing, **2** is *some* anti-aliasing, **4** is *full* anti-aliasing (default is **4**). See also **-g** and **-l**.
- u <user>**
Search *user*'s home directory for *peps.mime*. See the **MIME** section for details. This option has no effect on **xpeps**.
- v <resolution>**
Vertical resolution in ppi (default **100**). Overrides **-q** and **-Q**, as well as the vertical resolution set by **-r**.
- v** Stop processing the command line, print **peps** version and exit. Unlike the option above, this one takes no parameters, so **peps** can distinguish between the setting of the vertical resolution and the request to print its version.
- w <width>**
The output will be *width* pixels wide. This overrides the horizontal resolution set by the **-h** or **-r** switches and discards any overrides of horizontal resolution made by the **-Q** or **-W** switches. By default the width of the bitmap is determined mathematically from the horizontal resolution and the width of the bounding box. With this switch the horizontal resolution is calculated from the *width* in pixels and the width of the bounding box. This switch does not affect the height of the bitmap or the vertical resolution set by the **-q**, **-Q**, **-r** or **-v** switches, but it does replace any override of the vertical resolution made by the **-W** switch. In other words, **-W 500 -w 100** is the same as **-W 100**.
- W <width>**
Does everything the **-w** switch does, but also affects the height of the bitmap by setting the vertical resolution the same as the new horizontal resolution. As a result, the bitmap will be *width* pixels

wide while keeping its height/width ratio as determined by the bounding box. This switch overrides the **-h**, **-r**, **-v** and **-w** switches, and discards any overrides made by the **-q** and **-Q** switches.

-x <offset>

Pad the left side of the image by *offset* points. If *offset* is negative, a section of the left side of the image is removed. The value is in Postscript points (default **0**).

-X <offset>

Pad the right side of the image by *offset* points. If *offset* is negative, a section of the right side of the image is removed. The value is in Postscript points (default **0**).

-y <offset>

Pad the bottom of the image by *offset* points. If *offset* is negative, a section of the bottom of the image is removed. The value is in Postscript points (default **0**).

-Y <offset>

Pad the top of the image by *offset* points. If *offset* is negative, a section of the top of the image is removed. The value is in Postscript points (default **0**).

Effectively, the **-x**, **-X**, **-y**, and **-Y** switches allow you to add a frame around the image (its color will depend on the underlying Postscript code), or to crop the image.

-z Turn on *gzip* compression. By default on most systems it is on when the **-c** switch is used and the *HTTP_ACCEPT_ENCODING* environmental variable lists *gzip*, off otherwise. The system administrators can, however, compile **peps** with different defaults, so the **-z** switch lets you turn it on explicitly. Regardless of the switch, the *gzip* compression is turned off when an output file is selected with **-o**. Of course, you can use redirection if you want to create a compressed file:

```
peps -z -m image.eps > image.png.gz
```

By the way, setting the *GZIP* environmental variable to *-9* will improve the compression. See *gzip(1)* for more details. If you are using the Apache web server, you can set it in the *.htaccess* file:

```
SetEnv GZIP -9
```

This switch has no effect on **xpeps**.

-Z Turn off *gzip* compression. This switch has no effect on **xpeps**.

file Use *file* for input. By default, **peps** uses *stdin* for its input. You can override it by typing a file name on the command line, before or after any switches (of course, if a switch takes additional parameters, you need to type the file name after the parameters). **peps** will treat the file name as it treats any other option, so if you enter more than one file name, **peps** will use the last one.

That also means that **-i file** works as expected: *file* will be the input. Though **-i** tells **peps** to use *stdin*, the file name that follows will override it, making *file* the input.

In the unlikely case that the name of the *file* starts with a *-*, precede it with another *-*, otherwise **peps** will think it is a switch. So, if the input file name is **-weird-**, enter it on the command line like this:

```
--weird-
```

See <http://peps.redprince.net> for further details and examples.

MIME

peps can produce on the fly images on line from within CGI scripts. When used with the **-c** switch, it will write the necessary HTTP *Content-type* header to *stdout*. In order to do that, it must know the **MIME** type of its output. It will look for it in several places. It will quit looking when it finds the **MIME** type.

Peps will search the following places in the following order:

1. The file specified with the **-M** switch.
2. The file *peps.mime* located in a user directory. Which user? If the **-u** switch is used, in the home directory of the user specified by the switch. If the **-u** switch is not used, it will use the home directory of the user specified by the *PEPSMIME* environmental variable. If neither the **-u** switch is used nor does the *PEPSMIME* environmental variable exist, **peps** will look for *peps.mime* in the home directory of the effective user running **peps**.
3. The file *peps.mime* located in a system directory. By default that directory is */etc*, but the system administrator can compile **peps** to look for it in a different directory.
4. A list hardcoded in **peps**.

If **peps** does not find the MIME in any of the above, it will default to *image/x-device*, where *device* is the Ghostscript device used.

Because web servers normally run as *nobody* or a similar special user, it is necessary to inform **peps** which user's home directory to search for *peps.mime*. This can be done explicitly with **-u** switch. But it is so easy to forget to use the switch or perhaps to make a typing mistake. It is, therefore, a good idea to tell the web server to set the *PEPSMIME* environmental variable. On the Apache server, this can be accomplished by entering the following in the *.htaccess* file:

```
SetEnv PEPSMIME user
```

Substitute the actual user name for *user*. Remember to keep *peps.mime* readable by the process running the web server. This can be accomplished by issuing the following command:

```
chmod 644 peps.mime
```

It is only necessary to issue this command once.

SEE ALSO

gs(1), gzip(1)

HISTORY

Both **peps** and this manual were written by G. Adam Stanislav, <adam at redprince dot net>. Version 1.0 was written to meet the needs of the FreeBSD documentation project to convert Encapsulated Postscript to PNG bitmaps. Version 2.0 greatly expands on **peps** abilities, as it now can be used as a Unix filter, as an interpreter, and as a tool to produce on the fly images on line. **Xpeps** was added in version 2.0 as a debugging tool for **peps** CGI scripts.