NAME

potrace − transform bitmaps into vector graphics.

SYNOPSIS

potrace [*options***] [***filename...***]**

DESCRIPTION

potrace is a tool for tracing a bitmap, which means, transforming a bitmap into a smooth, scalable image. The input is a bitmap, which means, a pixel-based image composed of the two colors black and white only. The default output is an encapsulated PostScript file (EPS). A typical use is to create EPS files from scanned data, such as company or university logos, handwritten notes, etc. The resulting image is not "jaggy" like a bitmap, but smooth. It can then be rendered at any resolution.

potrace can read bitmaps in the following formats: PBM, PGM, PPM (collectively known as PNM, see pnm(5)), as well as BMP (Windows and OS/2 bitmap formats). The input image should only use the two colors black and white. If other pixel values appear in the input, they will be converted to black and white using a simple threshold method.

potrace can currently produce the following output formats: EPS, PostScript, PDF, SVG, DXF, GeoJSON, PGM, Gimppath, and XFig. Additional backends might be added in the future.

OPTIONS

The following options are supported. Dimensions (arguments of type *dim*) can have optional units, e.g. 6.5in, 15cm, 100pt. The default unit is inches (or centimeters, if this was configured at compile time, see COMPILE TIME CONFIGURATION below). For pixel-based output formats such as PGM, DXF, GeoJSON, and Gimppath, the default unit is pixels.

General options:

-h, --help	print help message and exit.
-v, --version	print version info and exit. This also shows the defaults that were compiled into this version of **potrace**.
-l, --license	print license info and exit.

Input/output options:

filename	Each file can hold an input image, or multiple concatenated input images. If filename arguments are given, then **potrace** will by default create one output file for each input filename given. The name of the output file is obtained from the input filename by changing its suffix according to the chosen backend. If changing the suffix is impossible because the names of the input and output files would be identical, then the output filename is created by adding the "-out" suffix to the name of the input file. If no filename arguments are given, then **potrace** acts as a filter, reading from standard input and writing to standard output. A filename of "-" may be given to specify reading from standard input.
-o *filename***, --output** *filename*	write output to this file. All output is directed to the specified file. If this option is used, then multiple input filenames are only allowed for multi-page backends (see BACKEND TYPES below). In this case, each input file may contain one or more bitmaps, and all the bitmaps from all the input files are processed and the output concatenated into a single file. A filename of "-" may be given to specify writing to standard output.
--	End of options. Any remaining arguments are interpreted as filenames. This also disables filter mode, even if no filenames are given. This is useful for shell scripts, because **potrace -- $FILENAMES** will behave correctly even for an empty list of filenames. However, **--** with an empty list of filenames is not permitted in conjunction with the **-o** option, because this would generate a document of zero pages, which none of the backends permit.

Backend selection:

For general information, see also BACKEND TYPES below.

-b *name***, --backend** *name*

Select backend by name, where *name* is one of eps, postscript, ps, pdf, pdfpage, svg, dxf, geojson, pgm, gimppath, xfig. Backend names can be abbreviated by a prefix as long as it is unambiguous. Backend names are case insensitive.

-e, --eps, -b eps, --backend eps

EPS backend (default). The output is an encapsulated PostScript file. This is a single-page, variable-sized backend.

-p, --postscript, -b ps, --backend ps

PostScript backend. The output is a PostScript file. This is a multi-page, fixed-size backend. If the input consists of multiple bitmaps, they are each rendered on a separate page.

-b pdf, --backend pdf

PDF backend. The output is a file in the Portable Document Format. If the input consists of multiple bitmaps, they are each rendered on a separate page. This is a multi-page, variable-sized, backend.

-b pdfpage, --backend pdfpage

The PDFPage backend is like the PDF backend, except that it is fixed-size like the PostScript backend.

-s, --svg, -b svg, --backend svg

SVG backend. The output is a Scalable Vector Graphics (SVG) file. This is a single-page, variable-sized backend.

-b dxf, --backend dxf

DXF backend. The output is a file in the Drawing Interchange Format (DXF). In this backend, all Bezier curves are approximated by piecewise circular arcs; this is suitable for processing in CAD software or for machining applications using CNC tools. This is a single-page, variable-sized, pixel-based backend. The **-u** option has no effect for this backend.

-b geojson, --backend geojson

GeoJSON backend. The output is a file in the format used by some applications processing geographical data. In this backend, all Bezier curves are approximated by 8 straight line segments. This is a single-page, variable-sized, pixel-based backend. The **-u** option has no effect for this backend.

-g, --pgm, -b pgm, --backend pgm

PGM backend. The output is a portable greymap (PGM) file. It is a convenient backend for antialiasing a bitmap image. This is a multi-page, variable-sized, pixel-based backend. If the input consists of more than one image, the images are concatenated in the output.

-b gimppath, --backend gimppath

Gimppath backend. This backend produces output suitable to be imported as a path by the GNU Image Manipulation Program (Gimp) (in the Layers, Channels & Paths dialog, select Paths, then right-click and select Import Path). The output is actually an SVG file. The differences to the SVG backend are: the **--opaque** option has no effect, the **--flat** option is always on, and the dimensions are pixel-based. This is a single-page, variable-sized, pixel-based backend.

-b xfig, --backend xfig

XFig backend. The output is a file in the XFig format. Note that XFig uses X-splines instead of Bezier curves, thus it is not possible to translate the output of **potrace** into the XFig format with absolute accuracy. This backend does a reasonable approximation using two control points for each Bezier curve segment. The **-u** option has no effect for this backend, because control points are always rounded to the nearest 1/1200 of an inch in XFig. Curve optimization is disabled. Implies **--opaque**.

Algorithm options:

For more detailed information on these options, see TECHNICAL DOCUMENTATION below.

-z *policy***, --turnpolicy** *policy*

specify how to resolve ambiguities in path decomposition. Must be one of black, white, right, left, minority, majority, or random. Default is minority. Turn policies can be abbreviated by an unambigous prefix, e.g., one can specify min instead of minority.

-t *n***, --turdsize** *n* suppress speckles of up to this many pixels.

-a *n***, --alphamax** *n*

set the corner threshold parameter. The default value is 1. The smaller this value, the more sharp corners will be produced. If this parameter is 0, then no smoothing will be performed and the output is a polygon. If this parameter is greater than 4/3, then all corners are suppressed and the output is completely smooth.

-n, --longcurve turn off curve optimization. Normally **potrace** tries to join adjacent Bezier curve segments when this is possible. This option disables this behavior, resulting in a larger file size.

-O *n***, --opttolerance** *n*

set the curve optimization tolerance. The default value is 0.2. Larger values allow more consecutive Bezier curve segments to be joined together in a single segment, at the expense of accuracy.

-u *n***, --unit** *n* set output quantization. Coordinates in the output are rounded to 1/unit pixels. The default of 10 usually gives good results. For some of the debug modes, a value of 100 gives more accurate output. This option has no effect for the XFig backend, which always rasterizes to 1/1200 inch, or for the DXF backend. For the GeoJSON backend, this option is only a hint; the actual rounding may be more, but not less, accurate than specified.

-d *n***, --debug** *n* produce debugging output of type n. This has different effects for different backends. For the PostScript/EPS backends, the values n=1,2,3 illustrate the intermediate stages of the **potrace** algorithm.

Scaling and placement options:

-P *format***, --pagesize** *format*

for fixed-size backends, set page size. The following formats can be specified: A4, A3, A5, B5, Letter, Legal, Tabloid, Statement, Executive, Folio, Quarto, 10x14. Format names are case insensitive. Also, an argument of the form *dim*x*dim* is accepted to specify arbitrary dimensions. The default page size is Letter (or A4, if this was configured at compile time, see COMPILE TIME CONFIGURATION below). Page format names can be abbreviated by a prefix as long as it is unambiguous. This option has no effect for variable-sized backends.

-W *dim***, --width** *dim*

set the width of output image (before any rotation and margins). If only one of width and height is specified, the other is adjusted accordingly so that the aspect ratio is preserved.

-H *dim***, --height** *dim*

set the height of output image. See **-W** for details.

-r *n***[x***n***], --resolution** *n***[x***n***]**

for dimension-based backends, set the resolution (in dpi). One inch in the output image corresponds to this many pixels in the input. Note that a larger value results in a smaller output image. It is possible to specify separate resolutions in the x and y directions by giving an argument of the form *n*x*n*. For variable-sized backends, the default resolution is 72dpi. For fixed-size backends, there is no default resolution; the image is by default scaled to fit on the page. This option has no effect for pixel-based backends. If **-W** or **-H** are specified, they take precedence.

-x *n***[x***n***], --scale** *n***[x***n***]**

for pixel-based backends, set the scaling factor. A value greater than 1 enlarges the output, a value between 0 and 1 makes the output smaller. The default is 1. It is possible to specify separate scaling factors for the x and y directions by giving an argument of the form *n*x*n*. This option has no effect for dimension-based backends. If **-W** or **-H** are specified, they take precedence.

-S *n***, --stretch** *n* set the aspect ratio. A value greater than 1 means the image will be stretched in the y direction. A value between 0 and 1 means the image will be compressed in the y direction.

-A *angle***, --rotate** *angle*

set the rotation angle (in degrees). The output will be rotated counterclockwise by this angle. This is useful for compensating for images that were scanned not quite upright.

-M *dim***, --margin** *dim*

set all four margins. The effect and default value of this option depend on the backend. For variable-sized backends, the margins will simply be added around the output image (or subtracted, in case of negative margins). The default margin for these backends is 0. For fixed-size backends, the margin settings can be used to control the placement of the image on the page. If only one of the left and right margin is given, the image will be placed this distance from the respective edge of the page, and similarly for top and bottom. If margins are given on opposite sides, the image is scaled to fit between these margins, unless the scaling is already determined explicitly by one or more of the **-W**, **-H**, **-r**, or **-x** options. By default, fixed-size backends use a non-zero margin whose width depends on the page size.

-L *dim***, --leftmargin** *dim*

set the left margin. See **-M** for details.

-R *dim***, --rightmargin** *dim*

set the right margin. See **-M** for details.

-T *dim***, --topmargin** *dim*

set the top margin. See **-M** for details.

-B *dim***, --bottommargin** *dim*

set the bottom margin. See **-M** for details.

--tight remove whitespace around the image before scaling and margins are applied. If this option is given, calculations of the width, height, and margins are based on the actual vector outline, rather than on the outer dimensions of the input pixmap, which is the default. In particular, the **--tight** option can be used to remove any existing margins from the input image. See the file placement.pdf for a more detailed illustration.

Color options:

These options are only supported by certain backends. The DXF and GeoJSON backends do not support color.

-C *#rrggbb***, --color** *#rrggbb*

set the foreground color of the output image. The default is black.

--fillcolor *#rrggbb*

set the fill color of the output image, i.e., the color of the "white" parts that are enclosed by "black" parts. The default is to leave these parts transparent. Implies **--opaque**. Please note that this option sets the background color; to set the foreground color, use **--color** instead.

--opaque fill in the white parts of the image opaquely, instead of leaving them transparent. This only applies to interior white parts, i.e., those that are enclosed inside a black outline. Opaqueness is always in effect for the XFig backend.

SVG options:

--group for SVG output, try to group related paths together. Each path is grouped together with all paths that are contained inside it, so that they can be moved around as a unit with an SVG editor. This makes coloring individual components slightly more cumbersome, and thus it is not the default.

--flat for SVG output, put the entire image into a single path. This makes it impossible to color the components individually, and thus it is not the default. But the resulting SVG file can be more easily imported by some applications such as Gimp. In fact, the Gimppath backend is a variation of the SVG backend with the **--flat** option and pixel-based scaling. The **--flat** option has no effect if **--opaque** has been selected.

PostScript/EPS/PDF options:

-c, --cleartext do not compress the output. This option disables the use of compression filters in the PostScript and PDF output. In the PostScript backend, if **-c** and **-q** are used together, the resulting output can be easily read by other programs or even by humans.

-2, --level2 use PostScript level 2 compression (default). The resulting file size is ca. 40% smaller than if the **-c** option is used.

-3, --level3 use PostScript level 3 compression, if available. This gives slightly smaller files than using **-2**, but the resulting files may not print on older PostScript level 2 printers. If support for PostScript level 3 compression has been disabled at compile time, a warning message is printed and level 2 compression is used instead.

-q, --longcoding turn off optimized numerical coding in PostScript output. Normally, **potrace** uses a very compact numerical format to represent Bezier curves in PostScript, taking advantage of existing redundancy in the curve parameters. This option disables this behavior, resulting in longer, but more readable output (particularly if the **-c** option is also used).

PGM options:

-G *n*, **--gamma** *n* set the gamma value for anti-aliasing (default is 2.2). Most computer displays do not render shades of grey linearly, i.e., a grey value of 0.5 is not displayed as being exactly halfway between black and white. The gamma parameter corrects for this, and therefore leads to nicer looking output. The default value of 2.2 is appropriate for most normal CRT displays.

Frontend options:

-k *n*, **--blacklevel** *n*

set the threshold level for converting input images to bitmaps. The **potrace** algorithm expects a bitmap, thus all pixels of the input images are converted to black or white before processing begins. Pixels whose brightness is less than *n* are converted to black, all other pixels to white. Here *n* is a number between 0 and 1. One case is treated specially: if the input is in an indexed color format with exactly 2 colors, then the blacklevel is ignored and the darker of the two colors is mapped to black.

Note: the method used by **potrace** for converting greymaps to bitmaps is very crude; much better results can be obtained if a separate program, such as **mkbitmap**(1), is used for this purpose. In particular, **mkbitmap**(1), which is distributed with **potrace**, has the ability to scale and interpolate the image before thresholding, which results in much better preservation of detail.

-i, --invert invert the input bitmap before processing.

Progress bar options:

--progress	display a progress bar for each bitmap that is processed. This is useful for interactive use. The default behavior is not to show any progress information.
--tty *mode*	set the terminal mode for progress bar rendering. Possible values are "vt100", which requires a vt100-compatible terminal, and "dumb", which uses only ASCII characters. The default is system dependent.

BACKEND TYPES

Backends can be classified in several ways, which affects the available command line options and their behavior:

Fixed-size or variable-sized:

For fixed-size backends, the size of the page is always the same (for example Letter or A4, as specified at compile time or by the **-P** option). By default, the image will be centered and scaled to fit the page size. For variable-size backends, the size of the page follows the size of the image. Currently the PostScript (PS), PDFPage, and XFig backends are fixed-size, and the remaining backends are variable-size.

Dimension-based or pixel-based:

In dimension-based backends, distances are measured in physical units such as inches or centimeters. In pixel-based backends, distances are measured in pixel units. The **-r** option only works for dimension-based backends, and the **-x** option only works for pixel-based backends. Currently, the DXF, PGM, Gimppath, and GeoJSON backends are pixel-based, and the remaining backends are dimension-based. Currently, all pixel-based backends are variable-sized.

Single-page or multi-page:

Single-page backends can only accept a single image. Multi-page backends can accept multiple images, typically one per page of output. Currently, the PostScript (PS), PDF, PDFPage, and PGM backends are multi-page, and the remaining backends are single-page. Note that multiple input images can be read in two ways: from multiple input files (with the **-o** option), or from a single input file that holds several concatenated images.

COMPILE TIME CONFIGURATION

Certain aspects of the behavior of **potrace** can be configured at compile time by passing the following options to the ./configure script.

--disable-zlib

compile **potrace** without the zlib compression library. This means PostScript level 3 compression will not be available.

--enable-metric

compile **potrace** with centimeters as the default unit instead of inches.

--enable-a4

compile **potrace** with A4 as the default page size.

EXIT STATUS

The exit status is 0 on successful completion, 1 if the command line was invalid, and 2 on any other error.

VERSION

1.11

AUTHOR

Peter Selinger <selinger at users.sourceforge.net>

Please see the file AUTHORS for a full list of other contributors.

TECHNICAL DOCUMENTATION

For a detailed technical description of the **potrace** algorithm, see the file potrace.pdf, which is available from the **potrace** web site. For information on the Potrace library API, see potracelib.pdf.

WEB SITE AND SUPPORT

The latest version of **potrace** is available from http://potrace.sourceforge.net/. This site also contains a list of frequently asked questions, as well as information on how to obtain support.

SEE ALSO

mkbitmap(1)

COPYRIGHT