

cvtool

A computer vision tool
version 0.2.3, January 16, 2008

Martin Lambers (marlam@marlam.de)

This manual was last updated January 16, 2008 for version 0.2.3 of cvtool.

Copyright © 2005, 2006, 2007, 2008 Martin Lambers

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

Table of Contents

1	Overview	1
1.1	Concept	1
1.2	Supported file types	1
1.2.1	NetPBM formats: ‘pnm’	1
1.2.2	PFS format: ‘pfs’	1
1.3	Output	1
1.4	Global options	2
1.5	Common parameters	2
1.5.1	Arrays and Matrices	2
1.5.2	Colors	2
1.6	Environment	3
1.7	Exit codes	3
2	Commands	4
2.1	Informational Commands	4
2.1.1	help	4
2.1.2	version	4
2.1.3	info	4
2.2	Stream Manipulation	4
2.2.1	combine	4
2.2.2	convert	5
2.2.3	create	5
2.2.4	foreach	5
2.2.5	merge	5
2.2.6	reverse	6
2.2.7	select	6
2.2.8	split	6
2.3	Resizing Frames	7
2.3.1	resize	7
2.3.2	cut	7
2.4	Transforming Frames	7
2.4.1	affine	7
2.4.2	flip	8
2.4.3	flop	8
2.4.4	rotate	8
2.4.5	scale	8
2.4.6	shear	8
2.5	Mixing Frames	9
2.5.1	blend	9
2.5.2	layer	9
2.5.3	mix	9
2.6	Color Manipulation	10
2.6.1	channelextract	10
2.6.2	channelcombine	10
2.6.3	color	10
2.6.4	gamma	10
2.6.5	invert	11
2.7	Drawing	11

2.7.1	draw	11
2.8	Filtering Frames.....	13
2.8.1	gauss.....	13
2.8.2	mean.....	13
2.8.3	median.....	13
2.8.4	min.....	13
2.8.5	max.....	14
2.8.6	convolve.....	14
2.8.7	laplace.....	14
2.8.8	unsharpmask.....	14
2.9	Detecting Image Features.....	15
2.9.1	edge.....	15
2.10	Comparing Frames.....	15
2.10.1	diff.....	15
2.11	High Dynamic Range (HDR) Images	15
2.11.1	tonemap.....	15
2.12	Miscellaneous.....	16
2.12.1	visualize.....	16
Appendix A	Command index.....	18
Appendix B	Copying Information.....	19
	GNU Free Documentation License.....	19
	GNU GPL.....	25

1 Overview

1.1 Concept

Cvtool is a filter that manipulates one or more images (called frames): it reads frames from standard input and writes the manipulated frames to standard output. It can read and write streams of NetPBM (pbm, pgm, ppm, pnm, pam) and PFS frames.

Cvtool integrates all its functionality into a single binary, and makes it available through commands such as `rotate`, `filter`, and others.

The following command scales a NetPBM frame by a factor of 3:

```
$ cvtool scale --factor 3.0 < input.ppm > output.ppm
```

This is how one would select a rectangle from an PFS frame stream:

```
$ cvtool cut --left 10 --top 10 --width 100 --height 100 \  
< input.pfs > output.pfs
```

`cvtool help` prints a list of available commands, and `cvtool help cmd` prints help for the command `cmd`.

1.2 Supported file types

Due to the use of OpenGL textures to store frame data, some limitations apply:

- The maximum frame size is limited by the OpenGL maximum texture size.
- The maximum number of channels in a frame is 4.

1.2.1 NetPBM formats: ‘pnm’

All NetPBM image formats (pbm, pgm, ppm, pnm, pam) are supported, except for their old "plain" variants. Multiple images in one file are supported. They may differ in size and type.

If the input images use more than 8 bit per channel, they are treated as floating point images.

1.2.2 PFS format: ‘pfs’

Cvtool supports the PFS format used by `pfstools`.

Currently, cvtool ignore channel tags. This will be fixed in a future version.

1.3 Output

Cvtool normally prints messages to `stderr`. It prepends messages with its name, the level of information, and the name of the command.

The level of information is `DBG` for debugging messages, `INF` for informational messages, `WRN` for warnings, `ERR` for error messages, and `REQ` for requested information. Normally, cvtool prints only messages of level `INF` or higher, but this can be changed with ‘`--quiet`’ and ‘`--verbose`’; see below.

Some commands, for example `info`, print special information messages that the user explicitly requests. Such special messages have the level `REQ`, and can usually be redirected using the ‘`--output`’ option. In this case, no additional information will be prepended to the messages.

The special filename `-` means standard output (`stdout`). Redirecting messages to `stdout` is only allowed when no frames are written to `stdout`.

1.4 Global options

`-q|--quiet`

Reduces the amount of output: only messages with level `WRN` and higher will be printed.

`-v|--verbose`

Increases the amount of output: all messages will be printed, even those with level `DBG`. This will include progress information in many cases, but much of the output is really only useful for debugging purposes.

1.5 Common parameters

1.5.1 Arrays and Matrices

Some commands need arrays of integer or floating point values as parameters. Matrices are treated as two-dimensional arrays. Higher dimensions are also possible.

All of these array types are treated the same: the first part of the argument determines the number of dimensions of the array and its size in each dimension. The second part lists all values, separated by commas.

If the command requests an array or matrix of fixed dimension and size (or of dimension 1 and arbitrary size), then the first part can be omitted: only the value list is necessary in this case.

Examples:

- An array with three integer values

```
3:1,1,1
```

- An array with five floating point values

```
5:1.2,1.3,0.7,0.5,0.0
1.2,1.3,0.7,0.5,0.0
```

- A 3x3 matrix array with integer values

```
3x3:1,2,3,4,5,6,7,8,9
```

- A three-dimensional array with floating point values

```
2x2x2:1.11,1.12,1.21,1.22,2.11,2.12,2.21,2.22
```

1.5.2 Colors

Colors can be given in one of three forms:

1. SVG color names

The SVG 1.1 specification of the W3C defines 147 color names. `Cvtool` accepts each of these names, case insensitively. The full list can be found here: <http://www.w3.org/TR/SVG11/types.html>.

2. Hex triplets

The RGB components of a color can be specified directly as a hex triplet: `0xrrggbb`. For example, `0x00ff00` is green, `0xffffffff` is white, and `0x000000` is black.

3. Decimal values

The RGB components of a color can be specified as decimal values, prepended with `r`, `g`, or `b`. For example, `g255` is green (the red and blue components default to zero), `r255g255b255` is white, and `r0g0b0` is black.

1.6 Environment

TMPDIR Directory to create temporary files in.

COLUMNS Cvttool tries to format its messages so that they do not use more than the given number of columns. If this variable is unset, a default of 80 will be used.

1.7 Exit codes

Cvttool returns 0 on success and 1 on error.

2 Commands

2.1 Informational Commands

2.1.1 help

`help [command]`

Print general or command specific help.

2.1.2 version

`version`

Print version information.

2.1.3 info

`info [-s|--statistics] [-S|--single] [-o|--output=file]`

Print information about frames in the input stream.

If ‘`--single`’ is used, the command exits after the first frame has been processed. If ‘`--statistics`’ is used, additional statistics about the frame contents are printed. The output can be redirected to a file or to standard output (-) using the ‘`--output`’ option.

The following information will be printed: STREAM (pfs or pnm), CHANNELS (0-4), FORMAT (luminance or color), TYPE (uint8 or float), WIDTH, HEIGHT.

Statistics are computed for each available channel *c*: CHc_MIN, CHc_MAX, CHc_MEAN, CHc_MEDIAN, CHc_STDDEVIATION.

Example:

```
$ cvtool info < file.pnm
cvtool: [REQ] info: STREAM=pnm CHANNELS=3 FORMAT=color TYPE=uint8 WIDTH=394 HEIGHT=454
$ eval 'cvtool info -o - < file.pnm'
$ echo $WIDTH
394
```

2.2 Stream Manipulation

2.2.1 combine

`combine [-m|--method=(lr|leftright)|(tb|topbottom)] [-j|--justify=(left|top)|center|(right|bottom)] [-c|--color=color] file...`

Combine the given files by placing the frames side by side (‘`leftright`’) or one below the other (‘`topbottom`’).

The default is ‘`leftright`’. If the frames have different sizes, then the smaller ones have to be aligned with the biggest one. The default is to center them. The remaining space will be filled with the given color; the default is black.

Example:

```
$ cvtool combine left.pnm right.pnm > lr.pnm
$ cvtool combine -m tb \
  <(cvtool combine a.pnm b.pnm) \
  <(cvtool combine c.pnm d.pnm) \
  > 2x2.pnm
```


2.2.2 convert

`convert [-t|--type=uint8|float] [-f|--format=lum|color]`

Converts the input frames to another type and format. The default is to keep the input type and format. The output will be PNM for type uint8 and PFS for type float.

Example:

```
$ cvtool convert -t float < in.pnm > out.pfs
```

2.2.3 create

`create [-t|--type=uint8|float] [-f|--format=lum|color] [-n|--n=n] -w|--width=w -h|--height=h [-c|--color=<color>]`

Create *n* (default 1) frames with the given format (default color) and the given type (default uint8). The frames will have the given width and height, and they will be filled with the given color (default black). The resulting stream type will be PNM if the type is uint8, and PFS otherwise.

Example:

```
$ cvtool create -t uint8 -f lum -w 720 -h 576 > out.pgm
$ cvtool create -t float -f color -w 10 -h 10 -c green > green.pfs
```

2.2.4 foreach

`foreach [-s|--shell=shell] [-n|--n=n] cmd`

Execute the given command for every frame.

The command is expected to read *n* frames from standard input (default is *n*=1), and write an arbitrary number (including zero) of frames to standard output. The original frame(s) that were given to the command are replaced by the output of the command. The frames that the command produces are converted to the format of the original frames. The `foreach` command replaces the following special strings in the command *cmd* before executing the command: `%N` (replaced with frame number), `%W` (replaced with frame width), and `%H` (replaced with frame height). If *n* is greater than 1, these values refer to the first frame that is piped to the command. The command *cmd* is executed by passing it to the system shell. The default is `/bin/sh -c` on most systems. This can be overridden with the `--shell` option. It expects a string with zero or one spaces: The first part of the string is the shell, the second part (if any) is the first option to the shell. The next option will then be the command to execute.

Example:

```
# Rotate a video. Resize after rotation to keep the original dimensions.
$ cvtool foreach 'cvtool rotate -a %N | cvtool resize -w 352 -h 240' \
  < video.pnm > rotating-video.pnm
```

2.2.5 merge

`merge [-s|--shuffle] [-o|--output=file] file...`

Merges files into one stream, in the given order.

If `--shuffle` is used, the order will be randomized. The file names will be printed to `stderr` in the order they are merged. If `--output` is used, the file names will be written to the given file instead.

Example:

```
$ ls
frame000.pnm frame001.pnm frame002.pnm
$ cvtool merge frame*.pnm > video.pnm
```

2.2.6 reverse

reverse

Reverses the order of the frames in the stream.

This requires a temporary file that is big enough to hold the complete input stream.

Example:

```
$ cvtool reverse < video.pnm > oediv.pnm
```

2.2.7 select

select [-d|--drop] [-f|--fps=*fps*] *range*...

Selects frames from a stream.

By default, frames in the given ranges are kept and all others dropped. With **--drop**, frames in the given ranges are dropped and all others kept.

A range must be of the following form: *l-h* (from *l* to *h*), *-h* (from beginning to *h*), *l-* (from *l* to end), *l* (only *l*), or *-* (everything). Each start and end point can be a frame number (counting from 0) or a time in the format [hours:]minutes:seconds[fraction]. In short: if it contains a colon, it's a time. Time ranges can only be used if the '**--fps**' option is used to specify the number of frames per second.

IMPORTANT: If you use frame number ranges, the high frame number is inclusive: the frame with this number will be dropped/kept. If you use time ranges, the high time is exclusive and marks the first frame that will not be dropped/kept.

Example:

```
# Drop the frames 0 to 124 from the stream (with a framerate of 25 fps,
# these are the first five seconds).
$ cvtool select --drop 0-124 < in.pnm > out.pnm
```

```
# Drop the first 5 seconds of the stream (with a framerate of 25 fps,
# these are the frames 0 to 124. The frame at 0:05, with the frame
# number 125, will be the first that is kept!)
$ cvtool select --fps 25 --drop 0:00-0:05 < in.pnm > out.pnm
```

```
# Keep the second 5-minutes-block and drop all the rest. Both
# commands are equivalent.
$ cvtool select --fps 25 5:00-10:00 < in.pnm > out.pnm
$ cvtool select --fps 25 --drop -5:00 10:00- < in.pnm > out.pnm
```

2.2.8 split

split [-n|--n=*n*] [-t|--template=*template*] [-b|--backwards] [-s|--start=*i*]

Split the input stream into multiple files.

Each new files contains *n* frames (default is *n*=1). The filename will be generated from the template: the template must contain exactly one appearance of the character %. This character must be followed by one of the digits 1 through 9. The digit must be followed by the uppercase character N. This special string *%xN* will be replaced by the number of the first frame of the stream contained in this file. The number will be left-padded with zeros until its width is at least *x* characters. The default template is 'frame-*%6N*'.

A start number *i* for the first frame can be given, and the frames can be counted backwards. If the frames are counted backwards, a start number is required, because negative frame numbers are not accepted.

Example:

```
$ cvtool split -t frame%3N.pnm < ../video.pnm
$ ls
frame000.pnm frame001.pnm frame002.pnm
$ cvtool split -s 99 -b -t img%2N.pnm < ../video.pnm
$ ls
img99.pnm img98.pnm img97.pnm
```

2.3 Resizing Frames

2.3.1 resize

```
resize -w|--width=w -h|--height=h [-x|--x-offset=x] [-y|--y-offset=y]
[-c|--color=color]
```

Resize the frames to the given new width and height.

Place the original frame contents at the position (x,y) relative to the new frame (these offsets may be negative). If no or an incomplete position is given, compute the missing part(s) so that the old contents are centered on the new frame. Fill holes that might result with the given color (default is black).

Example:

```
# Add a green border of 10 pixels to a 352x240 frame
$ cvtool resize -w 372 -h 260 -c green < img.pnm > img2.pnm
```

2.3.2 cut

```
cut -l|--left=l -t|--top=t -w|--width=w -h|--height=h
```

Only let the given rectangle through; cut the rest of each frame.

Example:

```
$ cvtool cut -l 0 -t 0 -w 10 -h 10 < in.pnm > out.pnm
```

2.4 Transforming Frames

Most geometric transformation commands support the option ‘--interpolation’ to choose one of the following interpolation types:

- **none**: No interpolation / Nearest Neighbor.
- **bilinear**: Bilinear interpolation.
- **biquadratic**: Biquadratic interpolation.
- **bicubic**: Default bicubic interpolation (Mitchell-Netravali).
- **bicubic-b-spline**: Bicubic B-Spline interpolation.
- **bicubic-cr-spline**: Bicubic Catmull-Rom Spline interpolation.

The default interpolation type is **bilinear**.

2.4.1 affine

```
affine -m|--matrix=2x2-matrix [-c|--color=color] [-i|--interpolation=i]
```

Apply the affine transformation defined by the given matrix (4 floating point values separated by commas) to the frames. The frame dimensions will be adapted so that the resulting frame will fit. Possible holes will be filled with the given color; the default is black.

Example:

```
$ cvtool affine -m 2.0,0.1,0.75,1.0 < in.pnm > out.pnm
```

2.4.2 flip

flip

Flip frames (left/right).

Example:

```
$ cvtool flip < in.pnm > out.pnm
```

2.4.3 flop

flop

Flop frames (top/bottom).

Example:

```
$ cvtool flop < in.pnm > out.pnm
```

2.4.4 rotate

```
rotate -a|--angle=angle [-c|--color=color] [-i|--interpolation=i]
```

Rotate frames with the given angle (in degrees), counterclockwise.

The dimensions of the rotated frame will be big enough to hold all informations from the source. "Holes" will be filled with the given color; the default is black.

Example:

```
$ cvtool rotate -a -45 < in.pnm > out.pnm
```

2.4.5 scale

```
scale [-w|--width=w] [-h|--height=h] [-i|--interpolation=i]
```

```
scale -x|--factor-x=factor-x -y|--factor-y=factor-y [-i|--interpolation=i]
```

```
scale -f|--factor=factor [-i|--interpolation=i]
```

Scale frames to new size.

First form: Give new width and/or height. If one value is missing, it is computed from the other so that the aspect ratio remains the same.

Second form: Give scale factors for width and height.

Third form: Give one scale factor for both width and height.

Example:

```
# The following three commands do the same for a 400x200 frame:
$ cvtool scale -w 100 -h 50      < in.pnm > out.pnm
$ cvtool scale -x 0.25 -y 0.25  < in.pnm > out.pnm
$ cvtool scale -f 0.25          < in.pnm > out.pnm
```

2.4.6 shear

```
shear [-x|--shear-x=angle-x] [-y|--shear-y=angle-y] [-c|--color=color]
[-i|--interpolation=i]
```

Shear frames in horizontal and/or vertical direction.

The frames are sheared with the given angle(s) from (-90,90). Negative angles shear clockwise. "Holes" will be filled with the given color; the default is black.

Example:

```
$ cvtool shear -x 20 -y 10 < in.pnm > out.pnm
```

2.5 Mixing Frames

2.5.1 blend

```
blend -s|--source=file [-a|--alpha=file] [-S|--single] [-x|--x=x] [-y|--y=y]
```

Blends the source into the frame stream, using an alpha map.

With no alpha map, the source is simply copied into the frames. *x* and *y* specify the position that the source should be copied to. The default is (0,0). Positions outside of the frames are possible: parts of the source that do not fit into the frames will be ignored. When ‘--single’ is used, only the first frame of the source will be used; this frame will be copied into all frames of the stream.

Example:

```
$ cvtool blend --single -s logo.pnm -a logo-alpha.pgm -x 700 -y 0 \
  < video.pnm > video-with-logo.pnm
```

2.5.2 layer

```
layer -m|--mode=min|max|median|or|and|xor|diff|add|xadd|sub|xsub|mul|div
file...
```

Layers the frames from the given files on top of each other, using the given mode.

Layering will be done for each channel separately. The input frames may differ in size. In this case, they will be implicitly scaled to a common size. Graylevel frames have

The modes are as follows:

- **min**: Use minimum value.
- **max**: Use maximum value.
- **median**: Use median value.
- **or**: Bitwise or.
- **and**: Bitwise and.
- **xor**: Bitwise xor.
- **diff**: Use difference between maximum and minimum value.
- **add**: Use sum of values.
- **xadd**: Use sum of values. The ranges are transformed so that the results fit in [0,1]. Example for two layers: $X = (A/2) + (B/2)$.
- **sub**: Subtract values from the first value.
- **xsub**: Subtract values from the first value. The ranges are transformed so that the results fit in [0,1]. Example for two layers: $X = (A/2) - (B/2) + 1/2$.
- **mul**: Multiply values.
- **div**: Divide values.

Example:

```
$ cvtool layer --mode=or red.pnm green.pnm blue.pnm \
  > allchannels.pnm
```

2.5.3 mix

```
mix -w|--firstweight=fw... [-W|--lastweight=lw...] [-s|--steps=s] [-b|--bias=b]
file...
```

Mixes the given sources into a single stream using the given weights.

The default is to produce a single step, i.e. one output frame for each set of input frames. If more steps are requested, the weights are interpolated between the set of first weights and the

set of last weights. By default, this interpolation is done linearly, which corresponds to a bias setting of 0.5. Smaller bias values will give more attention to the first weights, larger values more to the last weights. The bias must be from (0,1).

The input frames may differ in size. In this case, they will be implicitly scaled to a common size.

Example:

```
$ cvtool mix --weight=1,1 black.pgm white.pgm > gray.pgm
$ cvtool mix -w 1,0 -W 0,1 -s 10 black.pgm white.pgm \
  > from-black-to-white.pgm
```

2.6 Color Manipulation

2.6.1 channelextract

`channelextract -c|--channel=0|1|2|3|r|g|b|lum`

Extract the given channel from the input. If ‘channel’ is 0, 1, 2, or 3, then the data is copied unmodified. If the ‘channel’ is r, g, b, or lum, then the input is first converted to the red, green, blue, or luminance form.

Example:

```
$ cvtool channelextract -c r < color.ppm > red.pgm
```

2.6.2 channelcombine

`channelcombine file0 [file1 [file2 [file3]]]`

Extract the first channel from the given files and combine them into multichannel output data.

Example:

```
$ cvtool channelcombine red.pgm green.pgm blue.pgm > rgb.ppm
```

2.6.3 color

`color [-h|--hue=h] [-s|--saturation=s] [-l|--lightness=l] [-c|--contrast=c]`

Color adjustment.

Hue, saturation, lightness, and contrast are manipulated in the HSL (Hue, Saturation, Lightness) color space. *h* is an additive constant to the hue angle, in degrees. *s*, *l*, *c* measure the relative change in saturation, lightness, contrast: -1 means the result will be zero, 0 means the result will be the same as the original, and +1 means that the result will be two times as high as the original. Values greater than +1 are possible. For example, *s* = -1 will convert the input frames to graylevels. See the [Wikipedia entry for HSL color space](#) for more information.

Example:

```
$ cvtool color -h 120 < red.pnm > green.pnm
$ cvtool color -h 120 < green.pnm > blue.pnm
$ cvtool color -h 120 < blue.pnm > red.pnm
$ cvtool color -s -1 < colored.pnm > gray.pnm
$ cvtool color -l +1 < dark.pnm > light.pnm
```

2.6.4 gamma

`gamma`

Gamma correction.

Example:

```
$ cvtool gamma -g 2.2 < dark.pnm > bright.pnm
```

2.6.5 invert

invert

Invert input frames.

Example:

```
$ cvtool invert < in.pnm > out.pnm
```

2.7 Drawing

2.7.1 draw

```
draw [-w|--width=width] [-d|--dash=dash-specification] [-l|--line-
cap=butt|round|square] [-L|--line-join=miter|round|bevel] [-s|--border-
style=none|color|pattern|multipattern|linear-gradient|radial-gradient] [-c|--
border-color=color] [-p|--border-pattern=file] [-g|--border-gradient=gradient-
specification] [-S|--fill-style=none|color|pattern|multipattern|linear-
gradient|radial-gradient] [-C|--fill-color=color] [-P|--fill-pattern=file]
[-G|--fill-gradient=gradient-specification] [-f|--font-family=font] [-
t|--font-slant=normal|italic|oblique] [-W|--font-weight=normal|bold]
[-F|--font-size=size|size-x,size-y] [-j|--justify-x=left|right|center]
[-J|--justify-y=bottom|top|center] [-a|--antialias] [-u|--unit] command...
```

Draw simple geometric forms, lines and curves, and/or text.

The style for the object lines and the filling can be specified separately; it is either ‘none’ (line/filling is not drawn), ‘color’ (solid color), ‘pattern’ (a pattern read from a file), ‘multipattern’ (a different pattern for each input frame, all read from a file), ‘linear-gradient’ (a linear gradient), or ‘radial-gradient’ (a radial gradient). The default is the solid color black for lines and no filling.

A linear gradient specification `x0,y0,color0,x1,y1,color1` defines a gradient along the line from start point `x0,y0` (with color `color0` to end point `x1,y1` (with color `color1`). Any number of additional color stops can be added by appending an offset value and its associated color to the gradient specification. The offsets must be between 0.0 and 1.0 and describe the position on the gradient line, where 0.0 is the start point and 1.0 is the end point.

A radial gradient specification `x0,y0,r0,color0,x1,y1,r1,color1` defines a gradient from the start circle `x0,y0,r0` with color `color0` to the end circle `x1,y1,r1` with color `color1`. Additional stops can be added in the same way as for linear gradients.

The style of lines can be further adjusted with the ‘--width’, ‘--dash’, ‘--line-cap’, and ‘--line-join’ options. The ‘--width’ option selects the line width; it is 2.0 by default. The ‘--dash’ takes a list of values that specify alternating lengths for “line on” and “line off” segments of a line. If only one value is given, these lengths are equal. The ‘--line-cap’ option selects the style of line and curve ends. The ‘--line-join’ options selects the style of the meeting point of two line or curve segments.

Antialiasing can be turned on (default) and off with ‘--antialias’.

If ‘--unit’ is given, then all coordinates and sizes on the command line refer to a frame of size 1x1. All values are then scaled so that they match the real frame dimensions. For example, the point (0.5,0.5) will always be in the middle of a frame, regardless of the frame dimensions.

Text is drawn relative to the current drawing position (previously set with `move_to`, for example). By default, the current drawing position sets the bottom left point of the first character of the text. This can be changed with the ‘--justify-x’ and ‘--justify-y’ options. The font family, slant, weight, and size can be chosen. Note that you may not get an error message if ‘--font-family’ fails to set the given font, because the underlying library may not

report this error. If you use two values for the font size, then the first applies to the horizontal direction and the second to the vertical direction, so that you can scale the font asymmetrically.

A drawing command consists of a command name and parameter sets that define one or more instances of the command.

Simple geometric forms:

`rectangle topleft-x,topleft-y,width,height [...]`

Draw a rectangle.

`circle center-x,center-y,radius [...]`

Draw a circle.

`ellipse rect-topleft-x,rect-topleft-y,width,height [...]`

Draw an ellipse in the given enclosing rectangle.

`arc center-x,center-y,radius,start-angle,stop-angle`

Draw a part of a circle, from the given start angle to the given stop angle.

Text:

`text string`

Print the string at the current drawing position.

Lines and curves:

`move_to x,y [...]`

Move current point.

`line_to x,y [...]`

Draw a line from the old current point to the new current point.

`curve_to x0,y0,x1,y1,x2,y2 [...]`

Draw a curve from the old current point to the new current point `x2,y2`, Using the control points `x0,y0` and `x1,y1`.

`rel_move_to dx,dy [...]`

Move the current point using relative coordinates.

`rel_line_to dx,dy [...]`

Draw a line using relative coordinates.

`rel_curve_to dx0,dy0,dx1,dy1,dx2,dy2 [...]`

Draw a curve using relative coordinates.

`close`

Close the current line/curve figure: draw a line from the current point to the start point of the figure and combine start and end point into one point.

Open lines and curves will automatically be closed when drawing geometric forms or text.

The draw command is only a simple interface to the excellent [CAIRO](#) graphics library. Much of the [CAIRO documentation](#) is useful for this command, too, especially the [FAQ](#).

Example:

```
# Draw two green circles with a line width of 5.
$ cvtool draw -w 5 -c green circle 50,50,40 50,50,20 \
  < blank.pnm > circle.pnm
# The same, but filled with linear gradient from red to yellow to blue.
$ cvtool draw -w 5 -S linear-gradient \
  -G 10,50,red,90,50,blue,0.5,yellow circle 50,50,40 50,50,20 \
  < blank.pnm > circle.pnm
# Display one video inside another video in the form of a circle.
$ cvtool draw -s none -S multipattern -P video2.pnm circle 50,50,40 \
  < video1.onm > out.pnm
```



```
# Print a string exactly centered in the middle of blank.pnm.
$ cvtool draw -u -f "Serif" -F 0.1,0.1 -j center -J center \
  move_to 0.5,0.5 text "Hello world" \
  < blank.pnm > text.pnm
```

2.8 Filtering Frames

2.8.1 gauss

```
filter gauss [-3|--3d] -k|--k=k
filter gauss [-3|--3d] -s|--sigma=s
filter gauss [-3|--3d] -x|--k-x=kx -y|--k-y=ky [-t|--k-t=kt]
filter gauss [-3|--3d] [-k|--k=k] [-x|--k-x=kx] [-y|--k-y=ky] [-t|--k-t=kt]
[-s|--sigma=s] [--sigma-x=sx] [--sigma-y=sy] [--sigma-t=st]
```

Filter frames with a Gauss filter, in 2D or 3D (with the third dimension being the time). The kernel size can be given for each dimension, or once for all. It will be $(2kx+1) \times (2ky+1) \times (2kt+1)$. Different values for each direction lead to asymmetric filtering. The gauss filter can be specified by the sigma value(s): the mask size will be computed so that roughly 95% of the mass lies within the resulting mask. It is also possible to specify both sigma and k.

Example:

```
$ cvtool gauss --3d -k 3 < video.pnm > smoothed-video.pnm
```

2.8.2 mean

```
filter mean [-3|--3d] -k|--k=k
filter mean [-3|--3d] -x|--k-x=kx -y|--k-y=ky [-t|--k-t=kt]
```

Filter frames with a Mean filter, in 2D or 3D (with the third dimension being the time). The kernel size can be given for each dimension, or once for all. It will be $(2kx+1) \times (2ky+1) \times (2kt+1)$. Different values for each direction lead to asymmetric filtering.

Example:

```
$ cvtool mean -k 2 < in.pnm > out.pnm
```

2.8.3 median

```
filter median [-a|--approximated] [-3|--3d] -k|--k=k
filter median [-a|--approximated] [-3|--3d] -x|--k-x=kx -y|--k-y=ky [-t|--k-
t=kt]
```

Filter frames with a Median filter, in 2D or 3D (with the third dimension being the time). The kernel size can be given for each dimension, or once for all. It will be $(2kx+1) \times (2ky+1) \times (2kt+1)$. Different values for each direction lead to asymmetric filtering.

If the *-approximated* option is given, then the median will be approximated. This helps to allow larger mask sizes.

Example:

```
$ cvtool median -a -k 2 < in.pnm > out.pnm
```

2.8.4 min

```
filter min [-3|--3d] -k|--k=k
filter min [-3|--3d] -x|--k-x=kx -y|--k-y=ky [-t|--k-t=kt]
```

Filter frames with a Minimum filter, in 2D or 3D (with the third dimension being the time). The kernel size can be given for each dimension, or once for all. It will be $(2kx+1) \times (2ky+1) \times (2kt+1)$. Different values for each direction lead to asymmetric filtering.

Example:

```
$ cvtool min -k 2 < in.pnm > out.pnm
```

2.8.5 max

```
filter max [-3|--3d] -k|--k=k
filter max [-3|--3d] -x|--k-x=kx -y|--k-y=ky [-t|--k-t=kt]
```

Filter frames with a Maximum filter, in 2D or 3D (with the third dimension being the time). The kernel size can be given for each dimension, or once for all. It will be $(2kx+1) \times (2ky+1) \times (2kt+1)$. Different values for each direction lead to asymmetric filtering.

Example:

```
$ cvtool max -k 2 < in.pnm > out.pnm
```

2.8.6 convolve

```
convolve -K|--kernel=K
convolve -X|--vector-x=X -Y|--vector-y=Y [-T|--vector-t=T]
```

Convolve frames with the given convolution kernel.

Both 2D and 3D kernels are accepted (the third dimension is the time). If the kernel is separable, the vectors that generate it can be given instead, to reduce computation costs. All kernel elements must be integers. The size of the kernel must be an odd number in each dimension.

Example:

```
# Both commands are equivalent to 2D smoothing with the
# mean filter with k=1:
$ cvtool convolve -K 3x3:1,1,1,1,1,1,1,1,1 < in.pnm > out.pnm
$ cvtool convolve -X 3:1,1,1 -Y 3:1,1,1 < in.pnm > out.pnm
```

2.8.7 laplace

```
laplace [-c|--c=c]
```

Sharpens the input frames using the Laplace operator.

The sharpness factor c must be greater than or equal to zero. Larger values increase the effect. The default is 0.5.

Example:

```
$ cvtool laplace -c 0.7 < smooth.pnm > sharp.pnm
```

2.8.8 unsharpmask

```
unsharpmask -u|--unsharp=file [-c|--c=c]
```

Sharpens the input frames using unsharp masking.

The unsharp version of the input frames must be given using the ‘--unsharp’ option. It can be produced using e.g. a 3x3 Gauss filter. The sharpness parameter c must be from (0.5, 1.0). The default is 0.7.

Example:

```
$ cvtool unsharpmask -u smoothsmooth.pnm -c 0.7 < smooth.pnm > sharp.pnm
```

2.9 Detecting Image Features

2.9.1 edge

edge sobel

edge canny -s|--sigma=*sigma* -l|--low=*l* -h|--high=*h*

Detect edges.

Sobel will generate graylevel frames: the brighter a point, the stronger the edge.

Canny will generate binary frames. The *sigma* parameter is for Gauss smoothing. *l* and *h* are used for Hysteresis thresholding; both must be from [0,1].

If the input is PFS, then the output will be PFS too and will contain both a channel containing the edge strengths and a channel containing the edge directions. If the input is PNM, then the output will be graylevel frames containing only the strength information.

Example:

```
$ cvtool edge sobel < in.pgm > gray-edges.pgm
$ cvtool edge canny -s 1.2 -l 4 -h 8 < in.pgm > bw-edges.pgm
```

2.10 Comparing Frames

2.10.1 diff

diff [-s|--statistics] [-o|--output=*file*] *file-1 file-2*

Shows the differences between the two sources.

The sources must have the same pixel type, width, and height. This command produces frames of the same dimensions and of the same pixel type. Each pixel will be the absolute value of the difference of the corresponding pixels in the two sources. For RGB frames, the values will be computed for each channel separately.

If ‘--statistics’ is used, the command will also compute the minimum, maximum, mean, and median error, and the standard deviation. For RGB frames, these values will be computed for each channel separately. For YUV frames, only the Y channel is considered. The output will be printed to `stderr`, unless it is redirected with the ‘--output’ option. If the output is redirected to `stdout` (-), then only the statistics and no frames will be written to `stdout`.

Example:

```
$ cvtool create -w 10 -h 10 -c r255g0b0 > red.pnm
$ cvtool create -w 10 -h 10 -c r0g255b0 > green.pnm
$ cvtool diff -s -o - red.pnm green.pnm
frame pair 0: minimum error      = 1.0000 1.0000 0.0000
frame pair 0: maximum error      = 1.0000 1.0000 0.0000
frame pair 0: median error       = 1.0000 1.0000 0.0000
frame pair 0: mean error         = 0.9600 0.9600 0.0000
frame pair 0: standard deviation = 0.1969 0.1969 0.0000
```

2.11 High Dynamic Range (HDR) Images

2.11.1 tonemap

```
tonemap -m|--method=schlick94 [--brightness=b]          tonemap -m|--method=tumblin99
[-l|--max-absolute-luminance=l] [--display-adaptation-level=d] [--
max-displayable-contrast=c]          tonemap -m|--method=drago03 [-l|--max-
absolute-luminance=l] [--max-display-luminance=d] [--bias=b]          tonemap
```

```

-m|--method=reinhard05 [--intensity=i] [--light-adaptation=l] [--chromatic-
adaptation=c] tonemap -m|--method=ashikhmin02 [-l|--max-absolute-luminance=l]
[--local-contrast=c] tonemap -m|--method=durand02 [-l|--max-absolute-
luminance=l] [--sigma-spatial=ss] [--sigma-color=sc] [--base-contrast=bc]
tonemap -m|--method=reinhard02 [--key-value=a] [--white=w] [--sharpness=s]
[--epsilon=e]

```

Tone map frames.

High dynamic range (HDR) frames are read from standard input, and low dynamic range (LDR) frames are written to standard output. See the original papers for a description of the parameters. For some methods, the results should be gamma corrected.

The default for the maximum absolute luminance is to get it from the file (if specified), or else 150.0.

The default for schlick94 is $b=100.0$.

The defaults for tumblin99 are $d=100.0$, $c=70.0$.

The defaults for drago03 are $d=200.0$, $b=0.85$.

The defaults for reinhard05 are $i=0.0$, $l=0.5$, $c=0.5$.

The default for ashikhmin02 is $c=0.5$.

The defaults for durand02 are $ss=0.3$, $sc=0.4$, $bc=2.0$.

The defaults for reinhard02 are $a=0.1$, $w=1.0$, $s=10.0$, $e=0.5$.

See also:

For general information:

E. Reinhard, G. Ward, S. Pattanaik, and P. Debevec. High Dynamic Range Imaging: Acquisition, Display and Image-based Lighting. *Morgan Kaufmann*, 2005.

For the **schlick94** method:

Section 7.2.9 of the HDRI book.

For the **tumblin99** method:

Section 7.2.2 of the HDRI book.

For the **drago03** method:

F. Drago, K. Myszkowski, T. Annen and N. Chiba. Adaptive Logarithmic Mapping For Displaying High Contrast Scenes. *Proc. Eurographics 2003*.

For the **durand02** method:

F. Durand and J. Dorsey. Fast Bilateral Filtering for the Display of High-Dynamic-Range Images. *Proc. ACM SIGGRAPH 2002*, pp. 257-266.

2.12 Miscellaneous

2.12.1 visualize

```

visualize scalar [-p|--pseudo-color] [-m|--min=m] [-M|--max=M] [-l|--log=base]
visualize vector2 -m|--mode=color
visualize vector2 -m|--mode=needle [-x|--sample-x=x] [-y|--sample-y=y]
[-X|--dist-x=dx] [-Y|--dist-y=dy] [-f|--factor=f]

```

visualize scalar: Visualizes scalar values by transforming values from $[m,M]$ to $[0,1]$ and writing the result as graylevel frames. M and m are automatically determined from the input if they are not given. By default, the transformation is linear. If ‘--log’ is given, then the transformation will use the logarithm with the given base. If ‘--pseudo-color’ is given, then pseudo colors are used instead of gray levels.

visualize vector2: Reads vector fields and visualizes them. Visualization as colors: Each of the x,y,z components, which range from -1 to 1, are transformed to R,G,B values that range

from 0 to 1. Visualization as needle diagrams: Every x -th vector in horizontal direction and every y -th vector in vertical direction will be represented by a needle. The needles will have a distance of dx pixels in horizontal and dy pixels in vertical direction. The needle length is the length of the vector after it was scaled with the factor f . The default values are $x=y=dx=dy=10$, $f=1.0$.

Appendix A Command index

A

affine..... 7

B

blend..... 9

C

channelcombine 10
 channelextract 10
 color..... 10
 combine..... 4
 convert..... 5
 convolve 14
 create..... 5
 cut 7

D

diff..... 15
 draw..... 11

E

edge..... 15

F

flip..... 8
 flop..... 8
 foreach..... 5

G

gamma..... 10
 gauss..... 13

H

help..... 4

I

info..... 4
 invert..... 11

L

laplace..... 14
 layer..... 9

M

max..... 14
 mean..... 13
 median..... 13
 merge..... 5
 min..... 13
 mix..... 9

R

resize..... 7
 reverse..... 6
 rotate..... 8

S

scale..... 8
 select..... 6
 shear..... 8
 split..... 6

T

tonemap..... 15

U

unsharpmask..... 14

V

version..... 4
 visualize..... 16

Appendix B Copying Information

GNU Free Documentation License

Version 1.2, November 2002

Copyright © 2000,2001,2002 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document *free* in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “Document”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “you”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “Modified Version” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “Secondary Section” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “Invariant Sections” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

A section “Entitled XYZ” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “Acknowledgements”, “Dedications”, “Endorsements”, or “History”.) To “Preserve the Title” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document’s license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both

covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its

Title Page, then add an item describing the Modified Version as stated in the previous sentence.

- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the “History” section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled “Acknowledgements” or “Dedications”, Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled “Endorsements”. Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled “Endorsements” or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements.”

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (C)  year  your name.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.2
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover
Texts.  A copy of the license is included in the section entitled ‘‘GNU
Free Documentation License’’.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with...Texts.” line with this:

```
with the Invariant Sections being list their titles, with
the Front-Cover Texts being list, and with the Back-Cover Texts
being list.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

GNU GPL

Version 2, June 1991

Copyright © 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation’s software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:
 - a. You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
 - b. You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
 - c. If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent

and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:
 - a. Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
 - b. Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
 - c. Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.
5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.
7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.
9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT

HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

```
one line to give the program's name and a brief idea of what it does.
Copyright (C) yyyy  name of author
```

```
This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.
```

```
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type 'show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type 'show c' for details.
```

The hypothetical commands ‘show w’ and ‘show c’ should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than ‘show w’ and ‘show c’; they could even be mouse-clicks or menu items—whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a “copyright disclaimer” for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
‘Gnomovision’ (which makes passes at compilers) written by James Hacker.
```

```
signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.