convert - convert an image or sequence of images

SYNOPSIS

convert [ options ... ] input_file output_file

DESCRIPTION

Convert converts an input file using one image format to an output file with a differing image format. In addition, various types of image processing can be performed on the converted image during the conversion process. Convert recognizes the image formats listed in Image Magick(1).

EXAMPLES

To make a thumbnail of a JPEG image, use:

    convert -size 120x120 cockatoo.jpg -resize 120x120
    +profile "*" thumbnail.jpg

In this example, '-size 120x120' gives a hint to the JPEG decoder that the image is going to be downscaled to 120x120, allowing it to run faster by avoiding returning a full-resolution image. The output image will be scaled so its largest dimension is 120 pixels. The that might be present in the input and aren’t needed in the thumbnail.

To convert a TIFF image to a PostScript A4 page with the image in the lower left-hand corner, use:

    convert -page A4+0+0 image.tif document.ps

To convert a raw Gray image with a 128 byte header to a portable graymap, use:

    convert -depth 8 -size 768x512+128 gray:raw
        image.pgm

To convert a Photo CD image to a TIFF image, use:

    convert -size 1536x1024 img0009.pcd image.tif
    convert img0009.pcd[4] image.tif

To annotate an image with blue text using font 12x24 at position (100,100), use:

    convert -font helvetica -fill blue
        -draw "text 100,100 Cockatoo"
        bird.jpg bird.miff

To tile a 640x480 image with a JPEG texture with bumps use:

    convert -size 640x480 tile:bumps.jpg tiled.png

To surround an icon with an ornamental border use:

    convert -mattecolor "#697B8F" -frame 6x6 bird.jpg
        icon.png
OPTIONS
Options are processed in command line order. Any option you specify on the command line remains in effect for the set of images that follows, until the set is terminated by the appearance of any option or -noop. Some options only affect the decoding of images and others only the encoding. The latter can appear after the final group of input images.

For a more detailed description of each option, see ImageMagick(1).

-adjoin join images into a single multi-image file
-affine <matrix> drawing transform matrix
-antialias remove pixel aliasing
-append append a set of images
-average average a set of images
-background <color> the background color
-blur <radius>x<sigma> blur the image with a Gaussian operator
-border <width>x<height> surround the image with a border of color
-bordercolor <color> the border color
-box <color> set the color of the annotation bounding box
-cache <threshold> megabytes of memory available to the pixel cache
-channel <type> the type of channel
-charcoal <factor> simulate a charcoal drawing
-chop <width>x<height>{+-}<x>{+-}<y>{%} remove pixels from the interior of an image
-clip apply the clipping path, if one is present
-coalesce merge a sequence of images
-colorize <value> colorize the image with the pen color
-colors <value> preferred number of colors in the image
-colorspace <value> the type of colorspace
-comment <string> annotate an image with a comment
-compose <operator> the type of image composition
-compress <type> the type of image compression
-contrast
enhance or reduce the image contrast

- **crop** `<width>x<height>/{+-}<x>{+-}<y>{%}`
  preferred size and location of the cropped image

- **cycle** `<amount>`
  displace image colormap by amount

- **debug** enable debug printout

- **deconstruct**
  break down an image sequence into constituent parts

- **delay** `<1/100ths of a second>`
  display the next image after pausing

- **density** `<width>x<height>`
  vertical and horizontal resolution in pixels of the image

- **depth** `<value>`
  depth of the image

- **despeckle**
  reduce the speckles within an image

- **display** `<host:display[.screen]>`
  specifies the X server to contact

- **dispose** `<method>`
  GIF disposal method

- **dither**
  apply Floyd/Steinberg error diffusion to the image

- **draw** `<string>`
  annotate an image with one or more graphic primitives

- **edge** `<radius>`
  detect edges within an image

- **emboss**
  emboss an image

- **encoding** `<type>`
  specify the text encoding

- **Endian** `<type>`
  specify endianness (MSB or LSB) of output image

- **enhance**
  apply a digital filter to enhance a noisy image

- **equalize**
  perform histogram equalization to the image

- **fill** `<color>`
  color to use when filling a graphic primitive

- **filter** `<type>`
  use this type of filter when resizing an image

- **flatten**
  flatten a sequence of images

- **flip**
  create a "mirror image"

- **flop**
  create a "mirror image"

- **font** `<name>`
  use this font when annotating the image with text

- **frame** `<width>x<height>+<outer bevel width>+<inner bevel width>`
  surround the image with an ornamental border

- **fuzz** `<distance>{%}`
  colors within this distance are considered equal

- **gamma** `<value>`
  level of gamma correction

- **Gaussian** `<radius>x<sigma>`
blur the image with a Gaussian operator

-geometry <width>x<height>{+-}<x>{+-}<y>{%}{!}{<}{>}
pREFERRED size and location of the Image window.

-gravity <type>
direction primitive gravitates to when annotating the image.

-help print usage instructions

-implode <factor>
implode image pixels about the center

-intent <type>
use this type of rendering intent when managing the image color

-interlace <type>
the type of interlacing scheme

-label <name>
assign a label to an image

-level <value>
adjust the level of image contrast

-list <type>
the type of list

-loop <iterations>
add Netscape loop extension to your GIF animation

-map <filename>
choose a particular set of colors from this image

-mask <filename>
Specify a clipping mask

-matte store matte channel if the image has one

-mattecolor <color>
specify the color to be used with the -frame option

-median <radius>
apply a median filter to the image

-modulate <value>
vary the brightness, saturation, and hue of an image

-monochrome transform the image to black and white

-morph <frames>
morphs an image sequence

-mosaic create a mosaic from an image sequence

-negate replace every pixel with its complementary color

-noise <radius|type>
add or reduce noise in an image

-noop NOOP (no option)

-normalize transform image to span the full range of color values

-opaque <color>
change this color to the pen color within the image

-page <width>x<height>{+-}<x>{+-}<y>{%}{!}{<}{>}
size and location of an image canvas

-paint <radius>
simulate an oil painting

-pen <color>
specify the pen color for drawing operations

-ping efficiently determine image characteristics
-**pointsize** `<value>`
pointsize of the PostScript, OPTION1, or TrueType font

-**preview** `<type>`
image preview type

-**process** `<command>`
process a sequence of images

-**profile** `<filename>`
add ICM, IPTC, or generic profile to image

-**quality** `<value>`
JPEG/MIFF/PNG compression level

-**raise** `<width>x<height>`
lighten or darken image edges

-**region** `<width>x<height>[+][-]<x>[+][-]<y>`
apply options to a portion of the image

-**resize** `<width>x<height>[@][!][<]/>`
resize an image

-**roll** `{+/-}<x>{+/-}<y>`
roll an image vertically or horizontally

-**rotate** `<degrees>{<}{>}`
apply Paeth image rotation to the image

-**sample** `<geometry>`
scale image with pixel sampling

-**sampling_factor** `<horizontal_factor>x<vertical_factor>`
sampling factors used by JPEG or MPEG-2 encoder and YUV decoder/encoder.

-**scale** `<geometry>`
scale the image.

-**scene** `<value>`
set scene number

-**seed** `<value>`
pseudo-random number generator seed value

-**segment** `<cluster threshold>x<smoothing threshold>`
segment an image

-**shade** `<azimuth>x<elevation>`
shade the image using a distant light source

-**sharpen** `<radius>x<sigma>`
sharpen the image

-**shave** `<width>x<height>`
shave pixels from the image edges

-**shear** `<x degrees>x<y degrees>`
shear the image along the X or Y axis

-**size** `<width>x<height>{+offset}`
width and height of the image

-**solarize** `<factor>`
negate all pixels above the threshold level

-**spread** `<amount>`
displace image pixels by a random amount

-**stroke** `<color>`
color to use when stroking a graphic primitive

-**strokewidth** `<value>`
set the stroke width

-**swirl** `<degrees>`
swirl image pixels about the center
User Commands

-**texture** <filename>
  name of texture to tile onto the image background

-**threshold** <value>
  threshold the image

-**tile** <filename>
  tile image when filling a graphic primitive

-**transform**
  transform the image

-**transparent** <color>
  make this color transparent within the image

-**treeedepth** <value>
  tree depth for the color reduction algorithm

-**trim**
  trim an image

-**type** <type>
  the image type

-**units** <type>
  the type of image resolution

-**unsharp** <radius>x<sigma>
  sharpen the image with an unsharp mask operator

-**use_pixmap**
  use the pixmap

-**verbose**
  print detailed information about the image

-**view** <string>
  FlashPix viewing parameters

-**wave** <amplitude>x<wavelength>
  alter an image along a sine wave

-**write** <filename>
  write an image sequence [convert, composite]

  For a more detailed description of each option, see ImageMagick(1).

SEE ALSO
display(1), animate(1), identify(1), ImageMagick(1), montage(1), mogrify(1), composite(1)

AUTHORS
  John Cristy, ImageMagick Studio LLC,
  Glenn Randers-Pehrson, ImageMagick Studio LLC.