

```
$ convert rose: -contrast -contrast rose_c2.png
```

**-contrast-stretch** *black-point*

**-contrast-stretch** *black-point*{*xwhite-point*}{%}

Increase the contrast in an image by *stretching* the range of intensity values.

While performing the stretch, black-out at most *black-point* pixels and white-out at most *white-point* pixels. Or, if percent is used, black-out at most *black-point* % pixels and white-out at most *white-point* % pixels.

Prior to ImageMagick 6.4.7-0, **-contrast-stretch** will black-out at most *black-point* pixels and white-out at most *total pixels minus white-point* pixels. Or, if percent is used, black-out at most *black-point* % pixels and white-out at most *100% minus white-point* % pixels.

Note that **-contrast-stretch 0** will modify the image such that the image's min and max values are stretched to 0 and *QuantumRange*, respectively, without any loss of data due to burn-out or clipping at either end. This is not the same as **-normalize**, which is equivalent to **-contrast-stretch 0.15x0.05%** (or prior to ImageMagick 6.4.7-0, **-contrast-stretch 2%x99%**).

Internally operator works by creating a histogram bin, and then uses that bin to modify the image. As such some colors may be merged together when they originally fell into the same 'bin'.

All the channels are normalized in concert by the same amount so as to preserve color integrity, when the default **+channel** setting is in use. Specifying any other **-channel** setting will normalize the RGB channels independently.

See also **-auto-level** for a 'perfect' normalization of mathematical images.

This operator is under review for re-development.

**-convolve** *kernel*

Convolve an image with a user-supplied convolution kernel.

The *kernel* is a matrix specified as a comma-separated list of integers (with no spaces), ordered left-to right, starting with the top row. Presently, only odd-dimensioned kernels are supported, and therefore the number of entries in the specified *kernel* must be  $3^2=9$ ,  $5^2=25$ ,  $7^2=49$ , etc.

Note that the **lconvolve** operator supports the **lbias** setting. This option shifts the convolution so that positive and negative results are relative to a user-specified bias value. This is important for non-HDRI compilations of ImageMagick when dealing with convolutions that contain negative as well as positive values. This is especially the case with convolutions involving high pass filters or edge detection. Without an output bias, the negative values is clipped at zero.

When using an ImageMagick with the HDRI compile-time setting, **lbias** is not needed, as ImageMagick is able to store/handle any negative results without clipping to the color value range (0..QuantumRange). See the discussion on HDRI implementations of ImageMagick on the page [High Dynamic-Range Images](#). For more about HDRI go the ImageMagick [Usage](#) pages or this [Wikipedia](#) entry.

**-crop** *geometry*{*@*}{!}

Cut out one or more rectangular regions of the image.

See [Image Geometry](#) for complete details about the *geometry* argument.

The *width* and *height* of the *geometry* argument give the size of the image that remains after cropping, and *x* and *y* in the *offset* (if present) gives the location of the top left corner of the cropped image with respect to the original image. To specify the amount to be removed, use **-shave** instead.

If the *x* and *y* offsets are present, a single image is generated, consisting of the pixels from the cropping region. The offsets specify the location of the upper left corner of the cropping region measured downward and rightward with respect to the upper left corner of the image. If the **-gravity** option is present with *NorthEast*, *East*, or *SouthEast* gravity, it gives the distance leftward from the right edge of the image to the right edge of the cropping region. Similarly, if the **-gravity** option is present with *SouthWest*, *South*, or *SouthEast* gravity, the distance is measured upward between the bottom edges.

If the *x* and *y* offsets are omitted, a set of tiles of the specified geometry, covering the entire input image, is generated. The rightmost tiles and the bottom tiles are smaller if the specified geometry extends beyond the dimensions of the input image.

You can add the *@* to the geometry argument to equally divide the image into the number of tiles generated.

By adding an exclamation character flag to the geometry argument, the cropped images virtual canvas page size and offset is set as if the geometry argument was a viewport or window. This means the canvas page size is set to exactly the same size you specified, the image offset set relative top left corner of the region cropped.

If the cropped image 'missed' the actual image on its virtual canvas, a special single pixel transparent 'missed' image is returned, and a 'crop missed' warning given.

It might be necessary to **+repage** the image prior to cropping the image to ensure the crop coordinate frame is relocated to the upper-left corner of the visible image. Similarly you may want to use **+repage** after cropping to remove

the page offset that will be left behind. This is especially true when you are going to write to an image format such as PNG that supports an image offset.

### **-cycle** *amount*

displace image colormap by amount.

*Amount* defines the number of positions each colormap entry is shifted.

### **-debug** *events*

enable debug printout.

The *events* parameter specifies which events are to be logged. It can be either `None`, `All`, `Trace`, or a comma-separated list consisting of one or more of the following domains: `Accelerate`, `Annotate`, `Blob`, `Cache`, `Coder`, `Configure`, `Deprecate`, `Exception`, `Locale`, `Render`, `Resource`, `Security`, `TemporaryFile`, `Transform`, `X11`, or `User`.

For example, to log cache and blob events, use.

```
$ convert -debug "Cache,Blob" rose: rose.png
```

The `User` domain is normally empty, but developers can log user events in their private copy of ImageMagick.

To print the complete list of debug methods, use **-list debug**.

Use the **-log** option to specify the format for debugging output.

Use **+debug** to turn off all logging.

Debugging may also be set using the `MAGICK_DEBUG` environment variable. The allowed values for the `MAGICK_DEBUG` environment variable are the same as for the **-debug** option.

### **-decipher** *filename*

Decipher and restore pixels that were previously transformed by **-encipher**.

Get the passphrase from the file specified by *filename*.

For more information, see the webpage, [ImageMagick: Encipher or Decipher an Image](#).

### **-deconstruct**

find areas that has changed between images

Given a sequence of images all the same size, such as produced by **-coalesce**, replace the second and later images, with a smaller image of just the area that changed relative to the previous image.

The resulting sequence of images can be used to optimize an animation sequence, though will not work correctly for GIF animations when parts of the animation can go from opaque to transparent.

This option is actually equivalent to the **-layers** method `'compare-any'`.

### **-define** *key*{*=value*}...

add specific global settings generally used to control coders and image processing operations.

This option creates one or more definitions for coders and decoders to use while reading and writing image data. Definitions are generally used to control image file format coder modules, and image processing operations, beyond what is provided by normal means. Defined settings are listed in **-verbose** information ("`info:`" output format) as "Artifacts".

If *value* is missing for a definition, an empty-valued definition of a flag is created with that name. This used to control on/off options. Use **+define key** to remove definitions previously created. Use **+define ""** to remove all existing definitions.

The same 'artifact' settings can also be defined using the **-set "option:key" "value"** option, which also allows the use of **Format and Print Image Properties** in the defined value.

The *option* and *key* are case-independent (they are converted to lowercase for use within the decoders) while the *value* is case-dependent.

Such settings are global in scope, and affect all images and operations.

The following definitions are just some of the artifacts that are available:

#### **bmp:format=***value*

valid values are *bmp2*, *bmp3*, and *bmp4*. This option can be useful when the method of prepending



## About ImageMagick

### Binary Releases

[Unix](#)  
[Mac OS X](#)  
[iOS](#)  
[Windows](#)

### Command-line Tools

[Processing](#)  
[Options](#)  
[Usage](#)

### Program Interfaces

[MagickWand](#)  
[MagickCore](#)  
[PerlMagick](#)  
[Magick++](#)

### Install from Source

[Unix](#)  
[Windows](#)

### Resources

### Architecture

### Download

### Search

### Site Map

[Links](#)

### Support ImageMagick

[Development](#)

## ImageMagick Convert Command-Line Tool

[\[Example Usage • Option Summary\]](#)

Use the `convert` program to convert between image formats as well as resize an image, blur, crop, despeckle, dither, draw on, flip, join, re-sample, and much more. See [Command Line Processing](#) for advice on how to structure your `convert` command or see below for example usages of the command.

### Example Usage

We list a few examples of the `convert` command here to illustrate its usefulness and ease of use. To get started, lets convert an image in the JPEG format to PNG:

```
$ convert rose.jpg rose.png
```

Next, we reduce the image size before it is written to the PNG format:

```
$ convert rose.jpg -resize 50% rose.png
```



You can combine multiple image-processing operations to produce complex results:

```
$ convert -size 320x85 canvas:none -font Bookman-DemiItalic -pointsize 72 \  
-draw "text 25,60 'Magick'" -channel RGBA -blur 0x6 -fill darkred -stroke magenta \  
-draw "text 20,55 'Magick'" fuzzy-magick.png
```

# Magick

or here we resize an image with improved quality:

```
$ convert input.png -colorspace RGB +sigmoidal-contrast 11.6933 \  
-define filter:filter=Sinc -define filter>window=Jinc -define filter:lobes=3 \  
-resize 400% -sigmoidal-contrast 11.6933 -colorspace sRGB output.png
```

You can find additional examples of using `convert` in [Examples of ImageMagick Usage](#).

### Option Summary

The `convert` command recognizes these options. Click on an option to get more details about how that option works.

Option	Description
<a href="#">-adaptive-blur</a> <i>geometry</i>	adaptively blur pixels; decrease effect near edges
<a href="#">-adaptive-resize</a> <i>geometry</i>	adaptively resize image with data dependent triangulation.
<a href="#">-adaptive-sharpen</a> <i>geometry</i>	adaptively sharpen pixels; increase effect near edges
<a href="#">-adjoin</a>	join images into a single multi-image file
<a href="#">-affine</a> <i>matrix</i>	affine transform matrix

<b>-alpha</b>	on, activate, off, deactivate, set, opaque, copy', transparent, extract, background, or shape the alpha channel
<b>-annotate</b> <i>geometry text</i>	annotate the image with text
<b>-antialias</b>	remove pixel-aliasing
<b>-append</b>	append an image sequence
<b>-authenticate</b> <i>value</i>	decipher image with this password
<b>-auto-gamma</b>	automagically adjust gamma level of image
<b>-auto-level</b>	automagically adjust color levels of image
<b>-auto-orient</b>	automagically orient image
<b>-background</b> <i>color</i>	background color
<b>-bench</b> <i>iterations</i>	measure performance
<b>-bias</b> <i>value</i>	add bias when convolving an image
<b>-black-threshold</b> <i>value</i>	force all pixels below the threshold into black
<b>-blue-primary</b> <i>point</i>	chromaticity blue primary point
<b>-blue-shift</b> <i>factor</i>	simulate a scene at nighttime in the moonlight
<b>-blur</b> <i>geometry</i>	reduce image noise and reduce detail levels
<b>-border</b> <i>geometry</i>	surround image with a border of color
<b>-bordercolor</b> <i>color</i>	border color
<b>-brightness-contrast</b> <i>geometry</i>	improve brightness / contrast of the image
<b>-caption</b> <i>string</i>	assign a caption to an image
<b>-cdl</b> <i>filename</i>	color correct with a color decision list
<b>-channel</b> <i>type</i>	apply option to select image channels
<b>-charcoal</b> <i>radius</i>	simulate a charcoal drawing
<b>-chop</b> <i>geometry</i>	remove pixels from the image interior
<b>-clamp</b>	set each pixel whose value is below zero to zero and any the pixel whose value is above the quantum range to the quantum range (e.g. 65535) otherwise the pixel value remains unchanged.
<b>-clip</b>	clip along the first path from the 8BIM profile
<b>-clip-mask</b> <i>filename</i>	associate clip mask with the image
<b>-clip-path</b> <i>id</i>	clip along a named path from the 8BIM profile
<b>-clone</b> <i>index</i>	clone an image
<b>-clut</b>	apply a color lookup table to the image
<b>-contrast-stretch</b> <i>geometry</i>	improve the contrast in an image by 'stretching' the range of intensity value
<b>-coalesce</b>	merge a sequence of images
<b>-colorize</b> <i>value</i>	colorize the image with the fill color
<b>-color-matrix</b> <i>matrix</i>	apply color correction to the image.
<b>-colors</b> <i>value</i>	preferred number of colors in the image
<b>-colorspace</b> <i>type</i>	set image colorspace
<b>-combine</b>	combine a sequence of images
<b>-comment</b> <i>string</i>	annotate image with comment
<b>-compare</b>	compare image
<b>-</b> <i>complex operator</i>	perform complex mathematics on an image sequence
<b>-compose</b> <i>operator</i>	set image composite operator
<b>-composite</b>	composite image
<b>-compress</b> <i>type</i>	image compression type
<b>-contrast</b>	enhance or reduce the image contrast

<b>-convolve</b> <i>coefficients</i>	apply a convolution kernel to the image
<b>-crop</b> <i>geometry</i>	crop the image
<b>-cycle</b> <i>amount</i>	cycle the image colormap
<b>-decipher</b> <i>filename</i>	convert cipher pixels to plain
<b>-debug</b> <i>events</i>	display copious debugging information
<b>-define</b> <i>format:option</i>	define one or more image format options
<b>-deconstruct</b>	break down an image sequence into constituent parts
<b>-delay</b> <i>value</i>	display the next image after pausing
<b>-delete</b> <i>index</i>	delete the image from the image sequence
<b>-density</b> <i>geometry</i>	horizontal and vertical density of the image
<b>-depth</b> <i>value</i>	image depth
<b>-despeckle</b>	reduce the speckles within an image
<b>-direction</b> <i>type</i>	render text right-to-left or left-to-right
<b>-display</b> <i>server</i>	get image or font from this Xserver
<b>-dispose</b> <i>method</i>	layer disposal method
<b>-distribute-</b> <b>cache</b> <i>port</i>	launch a distributed pixel cache server
<b>-distort</b> <i>type</i> <i>coefficients</i>	distort image
<b>-dither</b> <i>method</i>	apply error diffusion to image
<b>-draw</b> <i>string</i>	annotate the image with a graphic primitive
<b>-duplicate</b> <i>count,indexes</i>	duplicate an image one or more times
<b>-edge</b> <i>radius</i>	apply a filter to detect edges in the image
<b>-emboss</b> <i>radius</i>	emboss an image
<b>-encipher</b> <i>filename</i>	convert plain pixels to cipher pixels
<b>-encoding</b> <i>type</i>	text encoding type
<b>-endian</b> <i>type</i>	endianness (MSB or LSB) of the image
<b>-enhance</b>	apply a digital filter to enhance a noisy image
<b>-equalize</b>	perform histogram equalization to an image
<b>-evaluate</b> <i>operator value</i>	evaluate an arithmetic, relational, or logical expression
<b>-evaluate-</b> <b>sequence</b> <i>operator</i>	evaluate an arithmetic, relational, or logical expression for an image sequence
<b>-extent</b> <i>geometry</i>	set the image size
<b>-extract</b> <i>geometry</i>	extract area from image
<b>-family</b> <i>name</i>	render text with this font family
<b>-features</b> <i>distance</i>	analyze image features (e.g. contract, correlations, etc.).
<b>-fft</b>	implements the discrete Fourier transform (DFT)
<b>-fill</b> <i>color</i>	color to use when filling a graphic primitive
<b>-filter</b> <i>type</i>	use this filter when resizing an image
<b>-flatten</b>	flatten a sequence of images
<b>-flip</b>	flip image in the vertical direction
<b>-floodfill</b> <i>geometry color</i>	floodfill the image with color
<b>-flop</b>	flop image in the horizontal direction
<b>-font</b> <i>name</i>	render text with this font
<b>-format</b> <i>string</i>	output formatted image characteristics
<b>-frame</b> <i>geometry</i>	surround image with an ornamental border
<b>-function</b> <i>name</i>	apply a function to the image
<b>-fuzz</b> <i>distance</i>	colors within this distance are considered equal
<b>-fx</b> <i>expression</i>	apply mathematical expression to an image channel(s)

<b>-gamma</b> <i>value</i>	level of gamma correction
<b>-gaussian-blur</b> <i>geometry</i>	reduce image noise and reduce detail levels
<b>-geometry</b> <i>geometry</i>	preferred size or location of the image
<b>-gravity</b> <i>type</i>	horizontal and vertical text placement
<b>-grayscale</b> <i>method</i>	convert image to grayscale
<b>-green-primary</b> <i>point</i>	chromaticity green primary point
<b>-help</b>	print program options
<b>-identify</b>	identify the format and characteristics of the image
<b>-ift</b>	implements the inverse discrete Fourier transform (DFT)
<b>-implode</b> <i>amount</i>	implode image pixels about the center
<b>-insert</b> <i>index</i>	insert last image into the image sequence
<b>-intensity</b> <i>method</i>	method to generate an intensity value from a pixel
<b>-intent</b> <i>type</i>	type of rendering intent when managing the image color
<b>-interlace</b> <i>type</i>	type of image interlacing scheme
<b>-interline-spacing</b> <i>value</i>	the space between two text lines
<b>-interpolate</b> <i>method</i>	pixel color interpolation method
<b>-interword-spacing</b> <i>value</i>	the space between two words
<b>-kerning</b> <i>value</i>	the space between two characters
<b>-label</b> <i>string</i>	assign a label to an image
<b>-lat</b> <i>geometry</i>	local adaptive thresholding
<b>-layers</b> <i>method</i>	optimize or compare image layers
<b>-level</b> <i>value</i>	adjust the level of image contrast
<b>-limit</b> <i>type value</i>	pixel cache resource limit
<b>-linear-stretch</b> <i>geometry</i>	linear with saturation histogram stretch
<b>-liquid-rescale</b> <i>geometry</i>	rescale image with seam-carving
<b>-list</b> <i>type</i>	Color, Configure, Delegate, Format, Magic, Module, Resource, or Type
<b>-log</b> <i>format</i>	format of debugging information
<b>-loop</b> <i>iterations</i>	add Netscape loop extension to your GIF animation
<b>-mask</b> <i>filename</i>	associate a mask with the image
<b>-mattecolor</b> <i>color</i>	frame color
<b>-median</b> <i>radius</i>	apply a median filter to the image
<b>-metric</b> <i>type</i>	measure differences between images with this metric
<b>-mode</b> <i>radius</i>	make each pixel the 'predominant color' of the neighborhood
<b>-modulate</b> <i>value</i>	vary the brightness, saturation, and hue
<b>-moments</b>	display image moments.
<b>-monitor</b>	monitor progress
<b>-monochrome</b>	transform image to black and white
<b>-morph</b> <i>value</i>	morph an image sequence
<b>-morphology</b> <i>method kernel</i>	apply a morphology method to the image
<b>-motion-blur</b> <i>geometry</i>	simulate motion blur
<b>-negate</b>	replace each pixel with its complementary color
<b>-noise</b> <i>radius</i>	add or reduce noise in an image
<b>-normalize</b>	transform image to span the full range of colors
<b>-opaque</b> <i>color</i>	change this color to the fill color
<b>-ordered-dither</b> <i>NxN</i>	ordered dither the image
<b>-orient</b> <i>type</i>	image orientation

<b>-page</b> <i>geometry</i>	size and location of an image canvas (setting)
<b>-paint</b> <i>radius</i>	simulate an oil painting
<b>-perceptible</b>	set each pixel whose value is less than <i>[epsilon]</i> to <i>-epsilon</i> or <i>epsilon</i> (whichever is closer) otherwise the pixel value remains unchanged.
<b>-ping</b>	efficiently determine image attributes
<b>-pointsize</b> <i>value</i>	font point size
<b>-polaroid</b> <i>angle</i>	simulate a Polaroid picture
<b>-poly</b> <i>terms</i>	build a polynomial from the image sequence and the corresponding terms (coefficients and degree pairs).
<b>-posterize</b> <i>levels</i>	reduce the image to a limited number of color levels
<b>-precision</b> <i>value</i>	set the maximum number of significant digits to be printed
<b>-preview</b> <i>type</i>	image preview type
<b>-print</b> <i>string</i>	interpret string and print to console
<b>-process</b> <i>image-filter</i>	process the image with a custom image filter
<b>-profile</b> <i>filename</i>	add, delete, or apply an image profile
<b>-quality</b> <i>value</i>	JPEG/MIFF/PNG compression level
<b>-quantize</b> <i>colospace</i>	reduce image colors in this colorspace
<b>-quiet</b>	suppress all warning messages
<b>-radial-blur</b> <i>angle</i>	radial blur the image
<b>-raise</b> <i>value</i>	lighten/darken image edges to create a 3-D effect
<b>-random-threshold</b> <i>low,high</i>	random threshold the image
<b>-red-primary</b> <i>point</i>	chromaticity red primary point
<b>-regard-warnings</b>	pay attention to warning messages.
<b>-region</b> <i>geometry</i>	apply options to a portion of the image
<b>-remap</b> <i>filename</i>	transform image colors to match this set of colors
<b>-render</b>	render vector graphics
<b>-repage</b> <i>geometry</i>	size and location of an image canvas
<b>-resample</b> <i>geometry</i>	change the resolution of an image
<b>-resize</b> <i>geometry</i>	resize the image
<b>-respect-parentheses</b>	settings remain in effect until parenthesis boundary.
<b>-roll</b> <i>geometry</i>	roll an image vertically or horizontally
<b>-rotate</b> <i>degrees</i>	apply Paeth rotation to the image
<b>-sample</b> <i>geometry</i>	scale image with pixel sampling
<b>-sampling-factor</b> <i>geometry</i>	horizontal and vertical sampling factor
<b>-scale</b> <i>geometry</i>	scale the image
<b>-scene</b> <i>value</i>	image scene number
<b>-seed</b> <i>value</i>	seed a new sequence of pseudo-random numbers
<b>-segment</b> <i>values</i>	segment an image
<b>-selective-blur</b> <i>geometry</i>	selectively blur pixels within a contrast threshold
<b>-separate</b>	separate an image channel into a grayscale image
<b>-sepia-tone</b> <i>threshold</i>	simulate a sepia-toned photo
<b>-set</b> <i>attribute value</i>	set an image attribute
<b>-shade</b> <i>degrees</i>	shade the image using a distant light source
<b>-shadow</b> <i>geometry</i>	simulate an image shadow
<b>-sharpen</b>	sharpen the image



<i>geometry</i>	
<b>-shave</b> <i>geometry</i>	shave pixels from the image edges
<b>-shear</b> <i>geometry</i>	slide one edge of the image along the X or Y axis
<b>-sigmoidal-contrast</b>	increase the contrast without saturating highlights or shadows
<i>geometry</i>	
<b>-smush</b> <i>offset</i>	smush an image sequence together
<b>-size</b> <i>geometry</i>	width and height of image
<b>-sketch</b>	simulate a pencil sketch
<i>geometry</i>	
<b>-solarize</b>	negate all pixels above the threshold level
<i>threshold</i>	
<b>-splice</b> <i>geometry</i>	splice the background color into the image
<b>-spread</b> <i>radius</i>	displace image pixels by a random amount
<b>-statistic</b> <i>type</i>	replace each pixel with corresponding statistic from the neighborhood
<i>geometry</i>	
<b>-strip</b>	strip image of all profiles and comments
<b>-stroke</b> <i>color</i>	graphic primitive stroke color
<b>-strokewidth</b>	graphic primitive stroke width
<i>value</i>	
<b>-stretch</b> <i>type</i>	render text with this font stretch
<b>-style</b> <i>type</i>	render text with this font style
<b>-swap</b> <i>indexes</i>	swap two images in the image sequence
<b>-swirl</b> <i>degrees</i>	swirl image pixels about the center
<b>-synchronize</b>	synchronize image to storage device
<b>-taint</b>	mark the image as modified
<b>-texture</b>	name of texture to tile onto the image background
<i>filename</i>	
<b>-threshold</b> <i>value</i>	threshold the image
<b>-thumbnail</b>	create a thumbnail of the image
<i>geometry</i>	
<b>-tile</b> <i>filename</i>	tile image when filling a graphic primitive
<b>-tile-offset</b>	set the image tile offset
<i>geometry</i>	
<b>-tint</b> <i>value</i>	tint the image with the fill color
<b>-transform</b>	affine transform image
<b>-transparent</b>	make this color transparent within the image
<i>color</i>	
<b>-transparent-color</b> <i>color</i>	transparent color
<b>-transpose</b>	flip image in the vertical direction and rotate 90 degrees
<b>-transverse</b>	flop image in the horizontal direction and rotate 270 degrees
<b>-treedepth</b> <i>value</i>	color tree depth
<b>-trim</b>	trim image edges
<b>-type</b> <i>type</i>	image type
<b>-undercolor</b>	annotation bounding box color
<i>color</i>	
<b>-unique-colors</b>	discard all but one of any pixel color.
<b>-units</b> <i>type</i>	the units of image resolution
<b>-unsharp</b>	sharpen the image
<i>geometry</i>	
<b>-verbose</b>	print detailed information about the image
<b>-version</b>	print version information
<b>-view</b>	FlashPix viewing transforms
<b>-vignette</b>	soften the edges of the image in vignette style
<i>geometry</i>	
<b>-virtual-pixel</b>	access method for pixels outside the boundaries of the image
<i>method</i>	
<b>-wave</b> <i>geometry</i>	alter an image along a sine wave
<b>-weight</b> <i>type</i>	render text with this font weight



---

© 1999-2014 ImageMagick Studio LLC

[Contact the Wizards](#)

Stack Overflow is a question and answer site for professional and enthusiast programmers. It's 100% free, no registration required.

[Take the 2-minute tour](#) ×

## Command line batch image cropping tool

10



5

is there any lightweight command line batch image cropping tool(Linux or Windows) which can handle a variety of the formats ?

[image](#) [image-manipulation](#) [crop](#) [os-agnostic](#)[share](#) | [improve this question](#)

edited Dec 12 '09 at 12:26

[pavium](#)  
7,343 ●2●13●27

asked Dec 12 '09 at 12:16

[iceman](#)  
1,375 ●5●29●61

Is there anything else about your system you'd like to tell us? for example, which OS? – [pavium](#) Dec 12 '09 at 12:20

Either Linux or Windows. I have both on my little netbook – [iceman](#) Dec 12 '09 at 12:22

[add comment](#)

### 4 Answers

[active](#) [oldest](#) [votes](#)

8



[Imagemagick](#) is what you want -- tried and true.

[share](#) | [improve this answer](#)

answered Dec 12 '09 at 12:19

[axel\\_c](#)  
3,880 ●12●30

Thought I would share this, I tried pasting the link in MSN messenger to a friend and no matter what it seems it's on a blacklist of MSN. I had to make tinyurl in order to share it. Strange. – [ScottN](#) Jul 9 '13 at 23:23

That is very weird. Does MSN Messenger still exist anyway? ;) – [axel\\_c](#) Jul 10 '13 at 16:12

Haha, well, I use Pidgin and it still uses the protocol that MSN had, I believe. Strange the domain would be blocked though. – [ScottN](#) Jul 10 '13 at 18:40

[add comment](#)

13



In Linux you can use

```
mogrify -crop {Width}x{Height}+{X}+{Y} +repage image.png
```

for CLI image manipulation

[share](#) | [improve this answer](#)


edited yesterday

[Ben Alpert](#)  
23.7k ●9●63●110

answered Feb 10 '12 at 22:44

[Ralph](#)  
131 ●1●2

1 Quick tutorial: [linuxtutorialblog.com/post/...](#) – [ReactiveRaven](#) May 22 '12 at 10:37

homebrew makes short work of this in Mac OS X also. `brew install imagemagick` Then to see your new goodies... `ls -lrt /usr/local/bin/` – [Richard Bronosky](#) Jul 17 '12 at 6:18 

Note that this doesn't *actually* crop the image *data*, it only blanks the area surrounding the cropping rectangle and writes the new dimensions+offsets into image metadata, but the 'physical' dimensions of the image will not be changed. Because of this, this method doesn't work well with GIFs for example. – [kralyk](#) Aug 30 '13 at 1:48

1 P.S. to *actually* crop the image in the sense in which this operation is commonly understood, use the `+repage` argument along with the crop operator. – [kralyk](#) Aug 30 '13 at 1:51 

[add comment](#)

9



Imagemagick's convert does the trick for me (and much more than cropping):

```
convert -crop +100+10 in.jpg out.jpg
```

crops 100 pixels off the left border, 10 pixels from the top.






tagged

[image](#) × 44836[crop](#) × 1166[image-manipulation](#) × 1083[os-agnostic](#) × 11asked **4 years ago**viewed **10705 times**active **yesterday**

### Related

- 1 [Operating system independent image addressing](#)
- 0 [Crop part of image in php and resize it](#)
- 3 [Image cropping tool for touch based phones using javascript](#)
- 3 [Batch tool to automaticly crop images?](#)
- 3 [Image processing batch tool to resample images with YCbCr=4:4:4](#)
- 1 [Crop and rotate images with JS and PHP](#)
- 1 [PHP - Crop an image, and center the resulting image onto a background, using GD](#)
- 1 [cropping tools for admin panel cms](#)
- 0 [Does anyone knows a image crop tool for iPad?](#)
- 0 [Get new crop coordinates after rotating a cropped image](#)

### Hot Network Questions


-  [Is it faster to count to the infinite going one by one or two by two?](#)
-  [Is consistency in citation style important?](#)
-  [Make beamer not look like beamer](#)
-  [Is there a way to set-up red green yellow lights that show up depending on what status the record is in?](#)
-  [How do I make players appreciate roleplaying fluff and their surroundings?](#)

convert -crop -100+0 in.jpg out.jpg

crops 100 pixels off the right, and so on. The Imagemagick website knows more:  
<http://www.imagemagick.org/Usage/crop/#crop>

share | improve this answer

answered Feb 12 '13 at 4:40

 **Klaus**  
155 ● 1 ● 5

add comment

▲

3

▼

I found `nconvert` pretty handy so far.

share | improve this answer

edited yesterday

 **a\_horse\_with\_no\_name**  
85k ● 8 ● 69 ● 122

answered Dec 12 '09 at 12:21





 **Joey**  
136k ● 17 ● 244 ● 347





2 Dead link. minChars – Tim Apr 19 '13 at 16:09



add comment

Your Answer

B I

Sign up or log in

 Sign up using Google

 Sign up using Facebook

 Sign up using Stack Exchange

Post as a guest


Name


Email


Post Your Answer


By posting your answer, you agree to the [privacy policy](#) and [terms of service](#).


Not the answer you're looking for? Browse other questions tagged `image` `image-manipulation` `crop` `os-agnostic` or [ask your own question](#).


-  Why do movie posters use credit fonts that are hard to read?


 What do Jews think angels look like?


 Enabling Steam Family Sharing


 "What's our vector, victor?"


 How to do collision detection for fluids in 3D?


 How to drop empty tables


 How can I add version control info (to footers say?)


 Leave certain part of a tikz-picture uncovered


 Do cans change the taste of beer?


 How to make a countdown?


 'He does more than just "fool" or "fooling" around'?

 Why is "except: pass" a bad programming practice?

 Connectedness of a section of an algebraic bundle

 Is it bad to start a sentence with "as"?

 Should I set the lens profile option in Lightroom if I already have the lens correction information on my Canon 6D?

 How to give a label to nodes which already contain a tikzpicture with nodes.

more hot questions

Game Development    Electrical Engineering    **more (14)**    **more (15)**    Arcade (gaming)    **more (15)**  
TeX- LaTeX    Android Enthusiasts    Bicycles    **more (7)**  
Information Security    Role-playing Games  
**more (21)**

site design / logo © 2014 stack exchange inc; user contributions licensed under cc by-sa 3.0 with attribution required  
rev 2014.2.5.1349