ImageMagick Utilities version 5.4.8

John Cristy
Glenn Randers-Pehrson

ImageMagick Studio LLC
http://www.imagemagick.org
Copyright

Copyright (C) 2002 ImageMagick Studio, a non-profit organization dedicated to making software imaging solutions freely available.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (“ImageMagick”), to deal in ImageMagick without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of ImageMagick, and to permit persons to whom the ImageMagick is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of ImageMagick.

The software is provided “as is”, without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and noninfringement. In no event shall ImageMagick Studio be liable for any claim, damages or other liability, whether in an action of contract, tort or otherwise, arising from, out of or in connection with ImageMagick or the use or other dealings in ImageMagick.

Except as contained in this notice, the name of the ImageMagick Studio shall not be used in advertising or otherwise to promote the sale, use or other dealings in ImageMagick without prior written authorization from the ImageMagick Studio.
## Contents

1 ImageMagick Utilities .................................................. 1
   1.1 NAME ................................................................. 1
   1.2 Synopsis ............................................................ 1
   1.3 Description .......................................................... 1
   1.4 Options ............................................................... 6
   1.5 Files and Formats .................................................... 41
   1.6 Environment .......................................................... 42
   1.7 Authors ............................................................... 42

2 Animate ................................................................. 43
   2.1 NAME ................................................................. 43
   2.2 Synopsis ............................................................ 43
   2.3 Description .......................................................... 43
   2.4 Examples ............................................................. 44
   2.5 Options ............................................................... 44
   2.6 Mouse Buttons ....................................................... 47
   2.7 Command Widget ..................................................... 48
   2.8 Keyboard Accelerators .............................................. 48
   2.9 X Resources .......................................................... 48
   2.10 Environment .......................................................... 50
   2.11 Acknowledgements .................................................. 50
   2.12 Authors ............................................................... 50

3 Composite ............................................................... 51
   3.1 NAME ................................................................. 51
   3.2 Synopsis ............................................................. 51
## Contents

3.3 Description ........................................ 51
3.4 Examples ........................................ 51
3.5 Options ........................................ 52
3.6 Environment .................................... 55
3.7 Authors ....................................... 55

4 Conjure ........................................ 56
   4.1 NAME ........................................ 56
   4.2 Synopsis ...................................... 56
   4.3 Description .................................. 56
   4.4 Options ...................................... 57
   4.5 Magick Scripting Language .................... 58
   4.6 Authors ..................................... 60

5 Convert ........................................ 61
   5.1 NAME ........................................ 61
   5.2 Synopsis ...................................... 61
   5.3 Description .................................. 61
   5.4 Examples ..................................... 61
   5.5 Options ...................................... 63
   5.6 Segmentation .................................. 70
   5.7 Environment .................................. 71
   5.8 Authors ..................................... 71

6 Display .......................................... 72
   6.1 NAME ........................................ 72
   6.2 Synopsis ...................................... 72
   6.3 Description .................................. 72
   6.4 Examples ..................................... 74
   6.5 Options ...................................... 75
   6.6 Mouse Buttons ................................ 79
   6.7 Command Widget ................................. 80
   6.8 Keyboard Accelerators ......................... 83
   6.9 X Resources .................................. 85
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.5</td>
<td>Options</td>
<td>113</td>
</tr>
<tr>
<td>8.6</td>
<td>Environment</td>
<td>116</td>
</tr>
<tr>
<td>8.7</td>
<td>Authors</td>
<td>116</td>
</tr>
<tr>
<td>9</td>
<td>Mogrify</td>
<td>117</td>
</tr>
<tr>
<td>9.1</td>
<td>NAME</td>
<td>117</td>
</tr>
<tr>
<td>9.2</td>
<td>Synopsis</td>
<td>117</td>
</tr>
<tr>
<td>9.3</td>
<td>Description</td>
<td>117</td>
</tr>
<tr>
<td>9.4</td>
<td>Examples</td>
<td>117</td>
</tr>
<tr>
<td>9.5</td>
<td>Options</td>
<td>118</td>
</tr>
<tr>
<td>9.6</td>
<td>Image Segmentation</td>
<td>125</td>
</tr>
<tr>
<td>9.7</td>
<td>Environment</td>
<td>125</td>
</tr>
<tr>
<td>9.8</td>
<td>Authors</td>
<td>125</td>
</tr>
<tr>
<td>10</td>
<td>Montage</td>
<td>126</td>
</tr>
<tr>
<td>10.1</td>
<td>NAME</td>
<td>126</td>
</tr>
<tr>
<td>10.2</td>
<td>Synopsis</td>
<td>126</td>
</tr>
<tr>
<td>10.3</td>
<td>Description</td>
<td>126</td>
</tr>
<tr>
<td>10.4</td>
<td>Examples</td>
<td>127</td>
</tr>
<tr>
<td>10.5</td>
<td>Options</td>
<td>128</td>
</tr>
<tr>
<td>10.6</td>
<td>X Resources</td>
<td>132</td>
</tr>
<tr>
<td>10.7</td>
<td>Environment</td>
<td>133</td>
</tr>
<tr>
<td>10.8</td>
<td>Acknowledgements</td>
<td>133</td>
</tr>
<tr>
<td>10.9</td>
<td>Authors</td>
<td>133</td>
</tr>
</tbody>
</table>
1

ImageMagick Utilities

1.1 NAME

ImageMagick - commandline utilities to create, edit, or convert images

1.2 Synopsis

animate [ options ... ] file [ [ options ... ] file ... ]
composite [ options ... ] image composite [ mask ] composited
conjure [ options ] script.msl [ [ options ] script.msl ]
convert [ [ options ... ] [ input file ... ] ... [ output file ] ]
display [ options ... ] file ... [ [ options ... ] file ... ]
identify file [ file ... ]
import [ options ... ] file
mogrify [ options ... ] file ...
montage [ options ... ] file [ [ options ... ] file ... ] output_file

1.3 Description

ImageMagick provides a suite of commandline utilities for creating, converting, editing, and displaying images:

Display is a machine architecture independent image processing and display program. It can display an image on any workstation display running an X server.
**Import** reads an image from any visible window on an X server and outputs it as an image file. You can capture a single window, the entire screen, or any rectangular portion of the screen.

**Montage** creates a composite by combining several separate images. The images are tiled on the composite image with the name of the image optionally appearing just below the individual tile.

**Convert** converts an input file using one image format to an output file with a differing image format.

**Mogrify** transforms an image or a sequence of images. These transforms include **image scaling, image rotation, color reduction**, and others. The transmogrified image **overwrites** the original image.

**Identify** describes the format and characteristics of one or more image files. It will also report if an image is incomplete or corrupt.

**Composite** composites images to create new images.

**Conjure** interprets and executes scripts in the Magick Scripting Language (MSL).

The **ImageMagick** utilities recognize the following image formats:

<table>
<thead>
<tr>
<th>Name</th>
<th>Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8BIM</td>
<td>*rw-</td>
<td>Photoshop resource format</td>
</tr>
<tr>
<td>AFM</td>
<td>*r-</td>
<td>TrueType font</td>
</tr>
<tr>
<td>APP1</td>
<td>*rw-</td>
<td>Photoshop resource format</td>
</tr>
<tr>
<td>ART</td>
<td>*r-</td>
<td>PF1: 1st Publisher</td>
</tr>
<tr>
<td>AVI</td>
<td>*r-</td>
<td>Audio/Visual Interleaved</td>
</tr>
<tr>
<td>AVS</td>
<td>*rw+</td>
<td>AVS X image</td>
</tr>
<tr>
<td>BIE</td>
<td>*rw-</td>
<td>Joint Bi-level Image experts Group interchange format</td>
</tr>
<tr>
<td>BMP</td>
<td>*rw+</td>
<td>Microsoft Windows bitmap image</td>
</tr>
<tr>
<td>CAPTION</td>
<td>*r+</td>
<td>Caption (requires separate size info)</td>
</tr>
<tr>
<td>CMYK</td>
<td>*rw-</td>
<td>Raw cyan, magenta, yellow, and black samples (8 or 16 bits, depending on the image depth)</td>
</tr>
<tr>
<td>CMYKA</td>
<td>*rw-</td>
<td>Raw cyan, magenta, yellow, black, and matte samples (8 or 16 bits, depending on the image depth)</td>
</tr>
<tr>
<td>CUT</td>
<td>*r-</td>
<td>DR Halo</td>
</tr>
<tr>
<td>DCM</td>
<td>*r-</td>
<td>Digital Imaging and Communications in Medicine image</td>
</tr>
<tr>
<td>DCX</td>
<td>*rw+</td>
<td>ZSoft IBM PC multi-page Paintbrush</td>
</tr>
<tr>
<td>DIB</td>
<td>*rw+</td>
<td>Microsoft Windows bitmap image</td>
</tr>
<tr>
<td>DPS</td>
<td>*r-</td>
<td>Display PostScript</td>
</tr>
<tr>
<td>DPX</td>
<td>*r-</td>
<td>Digital Moving Picture Exchange</td>
</tr>
<tr>
<td>EPDF</td>
<td>*rw-</td>
<td>Encapsulated Portable Document Format</td>
</tr>
<tr>
<td>EPI</td>
<td>*rw-</td>
<td>Adobe Encapsulated PostScript</td>
</tr>
</tbody>
</table>
Interchange format

**EPS** *rw-* Adobe Encapsulated PostScript

**EPS2** *rw-* Adobe Level II Encapsulated PostScript

**EPS3** *rw-* Adobe Level III Encapsulated PostScript

**EPSF** *rw-* Adobe Encapsulated PostScript

**EPSI** *rw-* Adobe Encapsulated PostScript

**EPT** *rw-* Adobe Encapsulated PostScript with TIFF preview

**FAX** *rw+* Group 3 FAX

**FILE** *rw-* Uniform Resource Locator

**FIT** *rw-* Flexible Image Transport System

**FPX** *rw-* FlashPix Format

**FTP** *rw-* Uniform Resource Locator

**G3** *rw-* Group 3 FAX

**GIF** *rw+* CompuServe graphics interchange format

**GIF87** *rw-* CompuServe graphics interchange format (version 87a)

**GRADIENT** *rw-* Gradual passing from one shade to another

**GRAY** *rw+* Raw gray samples (8 or 16 bits, depending on the image depth)

**H** *rw-* Internal format

**HDF** *rw+* Hierarchical Data Format

**HISTOGRAM** *rw-* Histogram of the image

**HTM** *rw-* Hypertext Markup Language and a client-side image map

**HTML** *rw-* Hypertext Markup Language and a client-side image map

**HTTP** *rw-* Uniform Resource Locator

**ICB** *rw+* Truevision Targa image

**ICM** *rw-* ICC Color Profile

**ICO** *rw-* Microsoft icon

**ICON** *rw-* Microsoft icon

**IMPLICIT** *rw-*

**IPTC** *rw-* IPTC Newsphoto

**JBG** *rw+* Joint Bi-level Image experts Group interchange format

**JBIG** *rw+* Joint Bi-level Image experts Group interchange format

**JP2** *rw-* JPEG-2000 JP2 File Format Syntax

**JPC** *rw-* JPEG-2000 Code Stream Syntax

**JEPG** *rw-* Joint Photographic Experts Group

**JPG** *rw-* Joint Photographic Experts Group
<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JFIF</td>
<td>Text image format</td>
</tr>
<tr>
<td>LOGO</td>
<td>ImageMagick Logo</td>
</tr>
<tr>
<td>M2V</td>
<td>MPEG-2 Video Stream</td>
</tr>
<tr>
<td>MAP</td>
<td>Colormap intensities (8 or 16 bits, depending on the image depth) and indices (8 or 16 bits, depending on whether colors exceeds 256).</td>
</tr>
<tr>
<td>MAT</td>
<td>MATLAB image format</td>
</tr>
<tr>
<td>MATTE</td>
<td>MATTE format</td>
</tr>
<tr>
<td>MIFF</td>
<td>Magick image format</td>
</tr>
<tr>
<td>MNG</td>
<td>Multiple-image Network Graphics</td>
</tr>
<tr>
<td>MONO</td>
<td>Bi-level bitmap in least-significant-byte-first order</td>
</tr>
<tr>
<td>MPC</td>
<td>Magick Persistent Cache image format</td>
</tr>
<tr>
<td>MPEG</td>
<td>MPEG-1 Video Stream</td>
</tr>
<tr>
<td>MPG</td>
<td>MPEG-1 Video Stream</td>
</tr>
<tr>
<td>MPR</td>
<td>Magick Persistent Registry</td>
</tr>
<tr>
<td>MSL</td>
<td>Magick Scripting Language</td>
</tr>
<tr>
<td>MTV</td>
<td>MTV Raytracing image format</td>
</tr>
<tr>
<td>MVG</td>
<td>Magick Vector Graphics</td>
</tr>
<tr>
<td>NETSCAPE</td>
<td>Netscape 216 color cube</td>
</tr>
<tr>
<td>NULL</td>
<td>Constant image of uniform color</td>
</tr>
<tr>
<td>OTB</td>
<td>On-the-air bitmap</td>
</tr>
<tr>
<td>P7</td>
<td>Xv thumbnail format</td>
</tr>
<tr>
<td>PAL</td>
<td>16bit/pixel interleaved YUV</td>
</tr>
<tr>
<td>PALM</td>
<td>Palm Pixmap format</td>
</tr>
<tr>
<td>PBM</td>
<td>Portable bitmap format (black and white)</td>
</tr>
<tr>
<td>PCD</td>
<td>Photo CD</td>
</tr>
<tr>
<td>PCDS</td>
<td>Photo CD</td>
</tr>
<tr>
<td>PCL</td>
<td>Page Control Language</td>
</tr>
<tr>
<td>PCT</td>
<td>Apple Macintosh QuickDraw/PICT</td>
</tr>
<tr>
<td>PCX</td>
<td>ZSoft IBM PC Paintbrush</td>
</tr>
<tr>
<td>PDB</td>
<td>Pilot Image Format</td>
</tr>
<tr>
<td>PDF</td>
<td>Portable Document Format</td>
</tr>
<tr>
<td>PFA</td>
<td>TrueType font</td>
</tr>
<tr>
<td>PFB</td>
<td>TrueType font</td>
</tr>
<tr>
<td>PFM</td>
<td>TrueType font</td>
</tr>
<tr>
<td>PGM</td>
<td>Portable graymap format (gray scale)</td>
</tr>
<tr>
<td>PICON</td>
<td>Personal Icon</td>
</tr>
<tr>
<td>PICT</td>
<td>Apple Macintosh QuickDraw/PICT</td>
</tr>
<tr>
<td>PIX</td>
<td>Alias/Wavefront RLE image format</td>
</tr>
<tr>
<td>PLASMA</td>
<td>Plasma fractal image</td>
</tr>
<tr>
<td>PM</td>
<td>X Windows system pixmap (color)</td>
</tr>
<tr>
<td>PNG</td>
<td>Portable Network Graphics</td>
</tr>
<tr>
<td>PNM</td>
<td>Portable anymap</td>
</tr>
<tr>
<td>Format</td>
<td>Read/Write</td>
</tr>
<tr>
<td>--------</td>
<td>------------</td>
</tr>
<tr>
<td>PPM</td>
<td>*rw+</td>
</tr>
<tr>
<td>PREVIEW</td>
<td>*-w-</td>
</tr>
<tr>
<td>PS</td>
<td>*rw+</td>
</tr>
<tr>
<td>PS2</td>
<td>*-w+</td>
</tr>
<tr>
<td>PS3</td>
<td>*-w+</td>
</tr>
<tr>
<td>PSD</td>
<td>*rw-</td>
</tr>
<tr>
<td>PTIF</td>
<td>*rw-</td>
</tr>
<tr>
<td>PWP</td>
<td>*r-</td>
</tr>
<tr>
<td>RAS</td>
<td>*rw+</td>
</tr>
<tr>
<td>RGB</td>
<td>*rw+</td>
</tr>
<tr>
<td>RGBA</td>
<td>*rw+</td>
</tr>
<tr>
<td>RLA</td>
<td>*r-</td>
</tr>
<tr>
<td>RLE</td>
<td>*r-</td>
</tr>
<tr>
<td>ROSE</td>
<td>*rw-</td>
</tr>
<tr>
<td>SCT</td>
<td>*r-</td>
</tr>
<tr>
<td>SFW</td>
<td>*r-</td>
</tr>
<tr>
<td>SGI</td>
<td>*rw+</td>
</tr>
<tr>
<td>SHTML</td>
<td>*-w-</td>
</tr>
<tr>
<td>STEGANO</td>
<td>*r-</td>
</tr>
<tr>
<td>SUN</td>
<td>*rw+</td>
</tr>
<tr>
<td>SVG</td>
<td>*rw+</td>
</tr>
<tr>
<td>TEXT</td>
<td>*rw+</td>
</tr>
<tr>
<td>TGA</td>
<td>*rw+</td>
</tr>
<tr>
<td>TIF</td>
<td>*rw+</td>
</tr>
<tr>
<td>TIFF</td>
<td>*rw+</td>
</tr>
<tr>
<td>TILE</td>
<td>*r-</td>
</tr>
<tr>
<td>TIM</td>
<td>*r-</td>
</tr>
<tr>
<td>TTF</td>
<td>*r-</td>
</tr>
<tr>
<td>TXT</td>
<td>*rw+</td>
</tr>
<tr>
<td>UIL</td>
<td>*-w-</td>
</tr>
<tr>
<td>UYVY</td>
<td>*rw-</td>
</tr>
<tr>
<td>VDA</td>
<td>*rw+</td>
</tr>
<tr>
<td>VICAR</td>
<td>*rw-</td>
</tr>
<tr>
<td>VID</td>
<td>*rw+</td>
</tr>
<tr>
<td>VIFF</td>
<td>*rw+</td>
</tr>
<tr>
<td>VST</td>
<td>*rw+</td>
</tr>
<tr>
<td>WBMP</td>
<td>*rw-</td>
</tr>
<tr>
<td>WMF</td>
<td>*r-</td>
</tr>
<tr>
<td>WPG</td>
<td>*r-</td>
</tr>
<tr>
<td>X</td>
<td>*rw-</td>
</tr>
<tr>
<td>XBM</td>
<td>*rw-</td>
</tr>
</tbody>
</table>
and white)

XC *r– Constant image uniform color
XCF *r– GIMP image
XML *r– Scalable Vector Graphics
XPM *rw– X Windows system pixmap (color)
XV *rw+ Khoros Visualization image
XWD *rw– X Windows system window dump (color)
YUV *rw– CCIR 601 4:1:1

Modes:

* Native blob support
r Read
w Write
+ Multi-image

Support for some of these formats require additional programs or libraries. README tells where to find this software.

Note, a format delineated with + means that if more than one image is specified, it is composited into a single multi-image file. Use +adjoin if you want a single image produced for each frame.

Your installation might not support all of the formats in the list. To get an up-to-date listing of the formats supported by your particular configuration, run "convert -list format".

Raw images are expected to have one byte per pixel unless ImageMagick is compiled in 16-bit mode. Here, the raw data is expected to be stored two bytes per pixel in most-significant-byte-first order. You can tell if ImageMagick was compiled in 16-bit mode by typing “convert” without any options, and looking for “Q:16” in the first line of output.

### 1.4 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.

This is a combined list of the commandline options used by the ImageMagick utilities (animate, composite, convert, display, identify, import, mogrify and montage).

In this document, angle brackets (“<>”) enclose variables and curly brackets (“{[]}”) enclose optional parameters. For example, “-fuzz <distance> { % }” means you can use the option "-fuzz 10" or "-fuzz 2%".
-adjoin  join images into a single multi-image file

By default, all images of an image sequence are stored in the same file. However, some formats (e.g. JPEG) do not support more than one image and are saved to separate files. Use +adjoin to force this behavior.

-affine <matrix>  drawing transform matrix

This option provides a transform matrix \( sx, rx, ry, sy, tx, ty \) for use by subsequent -draw or -transform options.

-antialias  remove pixel aliasing

By default antialiasing algorithms are used when drawing objects (e.g. lines) or rendering vector formats (e.g. WMF and Postscript). Use +antialias to disable use of antialiasing algorithms. Reasons to disable antialiasing include avoiding increasing colors in the image, or improving rendering speed.

-append  append a set of images

This option creates a single image where the images in the original set are stacked top-to-bottom. If they are not of the same width, any narrow images will be expanded to fit using the background color. Use +append to stack images left-to-right. The set of images is terminated by the appearance of any option. If the -append option appears after all of the input images, all images are appended.

-average  average a set of images

The set of images is terminated by the appearance of any option. If the -average option appears after all of the input images, all images are averaged.

-backdrop <color>  display the image centered on a backdrop.

This backdrop covers the entire workstation screen and is useful for hiding other X window activity while viewing the image. The color of the backdrop is specified as the background color. The color is specified using the format described in the “Color Names” section of X(1). Refer to “X Resources” in the manual page for display for details.

-background <color>  the background color

The color is specified using the format described in the “Color Names” section of X(1).
-blur <radius>x<sigma>  blur the image with a Gaussian operator

Blur with the given radius and standard deviation (sigma).

-border <width>x<height>  surround the image with a border of color

See -geometry for details about the geometry specification.

-bordercolor <color>  the border color

The color is specified using the format described in the “Color Names” section of X(1).

-borderwidth <geometry>  the border width

-box <color>  set the color of the annotation bounding box

The color is specified using the format described in the “Color Names” section of X(1).

See -draw for further details.

-cache <threshold>  megabytes of memory available to the pixel cache

Image pixels are stored in memory until 80 megabytes of memory have been consumed. Subsequent pixel operations are cached on disk. Operations to memory are significantly faster but if your computer does not have a sufficient amount of free memory you may want to adjust this threshold value.

-channel <type>  the type of channel

Choose from: Red, Green, Blue, Opacity, Cyan, Magenta, Yellow, or Black.

Use this option to extract a particular channel from the image. Matte, for example, is useful for extracting the opacity values from an image.

-charcoal <factor>  simulate a charcoal drawing

-chop <width>x<height>{+-}<x>{+-}<y>{%}  remove pixels from the interior of an image

Width and height give the number of columns and rows to remove, and x and y are offsets that give the location of the leftmost column and topmost row to remove.
The \( x \) offset normally specifies the leftmost column to remove. If the \texttt{-gravity} option is present with \texttt{NorthEast}, \texttt{East}, or \texttt{SouthEast} gravity, it gives the distance leftward from the right edge of the image to the rightmost column to remove. Similarly, the \( y \) offset normally specifies the topmost row to remove, but if the \texttt{-gravity} option is present with \texttt{SouthWest}, \texttt{South}, or \texttt{SouthEast} gravity, it specifies the distance upward from the bottom edge of the image to the bottom row to remove.

The \texttt{-chop} option removes entire rows and columns, and moves the remaining corner blocks leftward and upward to close the gaps.

\texttt{-clip} apply the clipping path, if one is present

If a clipping path is present, it will be applied to subsequent operations.

For example, if you type the following command:

\begin{verbatim}
convert -clip -negate cockatoo.tif negated.tif
\end{verbatim}

only the pixels within the clipping path are negated.

The \texttt{-clip} feature requires the XML library. If the XML library is not present, the option is ignored.

\texttt{-coalesce} merge a sequence of images

Each image \( N \) in the sequence after Image 0 is replaced with the image created by flattening images 0 through \( N \).

The set of images is terminated by the appearance of any option. If the \texttt{-coalesce} option appears after all of the input images, all images are coalesced.

\texttt{-colorize <value>} colorize the image with the pen color

Specify the amount of colorization as a percentage. You can apply separate colorization values to the red, green, and blue channels of the image with a colorization value list delineated with slashes (e.g. 0/0/50).

\texttt{-colormap <type>} define the colormap type

Choose between \texttt{shared} or \texttt{private}.

This option only applies when the default X server visual is \texttt{PseudoColor} or \texttt{GRAYScale}. Refer to \texttt{-visual} for more details. By default, a shared colormap is allocated. The image shares colors with other X clients. Some image colors could be approximated, therefore your image may look very different than intended. Choose \texttt{Private} and the image colors appear exactly as they are defined. However, other clients may go \texttt{technicolor} when the image colormap is installed.
-colors <value>  preferred number of colors in the image

The actual number of colors in the image may be less than your request, but
never more. Note, this is a color reduction option. Images with less unique colors
than specified with this option will have any duplicate or unused colors removed.
Refer to quantize for more details.

Note, options -dither, -colorspace, and -treedepth affect the color reduction
algorithm.

-colorspace <value>  the type of colorspace

Choices are: GRAY, OHTA, RGB, Transparent, XYZ, YCbCr, YIQ, YPbPr,
YUV, or CMYK.

Color reduction, by default, takes place in the RGB color space. Empirical ev-
dence suggests that distances in color spaces such as YUV or YIQ correspond
to perceptual color differences more closely than do distances in RGB space.
These color spaces may give better results when color reducing an image. Refer
to quantize for more details.

The Transparent color space behaves uniquely in that it preserves the matte
channel of the image if it exists.

The -colors or -monochrome option is required for this option to take effect.

-comment <string>  annotate an image with a comment

Use this option to assign a specific comment to the image. You can include
the image filename, type, width, height, or other image attribute by embedding
special format characters:

%b  file size
%c  comment
%d  directory
%e  filename extension
%f  filename
%h  height
%i  input filename
%k  number of unique colors
%l  label
%m  magick
%n  number of scenes
%o  output filename
%p  page number
%q  quantum depth
%s  scene number
%t  top of filename
%u  unique temporary filename
%w  width
%x  x resolution
%y  y resolution
%#  signature
\n  newline
\r  carriage return

For example,

    -comment "%m:%f %wx%h"

produces an image comment of **MIFF:bird.miff 512x480** for an image titled
**bird.miff** and whose width is 512 and height is 480.

If the first character of *string* is @, the image comment is read from a file titled
by the remaining characters in the string.

**-compose <operator>**  the type of image composition

[This option is not used by *convert* but this section is included because it de-
scribes the composite operators that are used by the -draw option of *convert*.]

By default, each of the composite image pixels are replaced by the corresponding
image tile pixel. You can choose an alternate composite operation:

    Over
    In
    Out
    Atop
    Xor
    Plus
    Minus
    Add
    Subtract
    Difference
    Multiply
    Bumpmap
    Copy
    CopyRed
    CopyGreen
    CopyBlue
    CopyOpacity

How each operator behaves is described below.
Over
The result will be the union of the two image shapes, with opaque areas of composite image obscuring image in the region of overlap.

In
The result is simply composite image cut by the shape of image. None of the image data of image will be in the result.

Out
The resulting image is composite image with the shape of image cut out.

Atop
The result is the same shape as image image, with composite image obscuring image where the image shapes overlap. Note this differs from over because the portion of composite image outside image’s shape does not appear in the result.

Xor
The result is the image data from both composite image and image that is outside the overlap region. The overlap region will be blank.

Plus
The result is just the sum of the image data. Output values are cropped to 255 (no overflow). This operation is independent of the matte channels.

Minus
The result of composite image - image, with underflow cropped to zero. The matte channel is ignored (set to 255, full coverage).

Add
The result of composite image + image, with overflow wrapping around (mod 256).

Subtract
The result of composite image - image, with underflow wrapping around (mod 256). The add and subtract operators can be used to perform reversible transformations.

Difference
The result of abs(composite image - image). This is useful for comparing two very similar images.

Multiply
The result of composite image * image. This is useful for the creation of drop-shadows.

Bumpmap
The result image shaded by composite image.

Copy
The resulting image is image replaced with composite image. Here the matte information is ignored.

CopyRed
The resulting image is the red layer in image replaced with the red layer in composite image. The other layers are copied untouched.

CopyGreen
The resulting image is the green layer in image replaced with the green layer in composite image. The other layers are copied untouched.
CopyBlue
The resulting image is the blue layer in *image* replaced with the blue layer in *composite image*. The other layers are copied untouched.

CopyOpacity
The resulting image is the matte layer in *image* replaced with the matte layer in *composite image*. The other layers are copied untouched.

The image compositor requires a matte, or alpha channel in the image for some operations. This extra channel usually defines a mask which represents a sort of a cookie-cutter for the image. This is the case when matte is 255 (full coverage) for pixels inside the shape, zero outside, and between zero and 255 on the boundary.

For certain operations, if *image* does not have a matte channel, it is initialized with 0 for any pixel matching in color to pixel location (0,0), otherwise 255 (to work properly borderwidth must be 0).

**-compress <type>**  the type of image compression

Choices are: None, BZip, Fax, Group4, JPEG, Lossless, LZW, RLE or Zip.

Specify +compress to store the binary image in an uncompressed format. The default is the compression type of the specified image file.

If LZW compression is specified but LZW compression has not been enabled, the image data will be written in an uncompressed LZW format that can be read by LZW decoders. This may result in larger-than-expected GIF files.

“Lossless” refers to lossless JPEG, which is only available if the JPEG library has been patched to support it.

Use the -quality option to set the compression level to be used by JPEG, PNG, MIFF, and MPEG encoders. Use the -sampling_factor option to set the sampling factor to be used by JPEG, MPEG, and YUV encoders for downsampling the chroma channels.

**-contrast** enhance or reduce the image contrast

This option enhances the intensity differences between the lighter and darker elements of the image. Use -contrast to enhance the image or +contrast to reduce the image contrast.

**-crop <width>x<height>[+-]<x>[+-]<y>[%]**  preferred size and location of the cropped image

See -geometry for details about the geometry specification.

The width and height give the size of the image that remains after cropping, and x and y are offsets that give 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.
To specify a percentage width or height to be removed instead, append %. For example to crop the image by ten percent (five percent on each side of the image), use `-crop 10%`.

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.

```
-cycle <amount>  displace image colormap by amount
                     Amount defines the number of positions each colormap entry is shifted.
```

```
-debug  enable debug printout
```

```
-deconstruct  break down an image sequence into constituent parts
                     The sequence of images is terminated by the appearance of any option. If the
                     -deconstruct option appears after all of the input images, all images are deconstructed.
```

```
-delay <1/100ths of a second>  display the next image after pausing
                     This option is useful for regulating the animation of image sequences Delay/100
                     seconds must expire before the display of the next image. The default is no delay
                     between each showing of the image sequence. The maximum delay is 65535.
                     You can specify a delay range (e.g. -delay 10-500) which sets the minimum and
                     maximum delay.
```

```
-density <width>x<height>  vertical and horizontal resolution in pixels of the
                     image
                     This option specifies an image density when decoding a PostScript or Portable
                     Document page. The default is 72 dots per inch in the horizontal and vertical
                     direction. This option is used in concert with -page.
```
-**depth <value>** depth of the image

This is the number of bits in a color sample within a pixel. The only acceptable values are 8 or 16. Use this option to specify the depth of raw images whose depth is unknown such as GRAY, RGB, or CMYK, or to change the depth of any image after it has been read.

-**descend** obtain image by descending window hierarchy

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

-**displace <horizontal scale> x <vertical scale>** shift image pixels as defined by a displacement map

With this option, composite image is used as a displacement map. Black, within the displacement map, is a maximum positive displacement. White is a maximum negative displacement and middle gray is neutral. The displacement is scaled to determine the pixel shift. By default, the displacement applies in both the horizontal and vertical directions. However, if you specify mask, composite image is the horizontal X displacement and mask the vertical Y displacement.

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

This option is used with convert for obtaining image or font from this X server. See X(1).

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

Here are the valid methods:

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No disposal specified.</td>
</tr>
<tr>
<td>1</td>
<td>Do not dispose between frames.</td>
</tr>
<tr>
<td>2</td>
<td>Overwrite frame with background color from header.</td>
</tr>
<tr>
<td>3</td>
<td>Overwrite with previous frame.</td>
</tr>
</tbody>
</table>

-**dissolve <percent>** dissolve an image into another by the given percent

The opacity of the composite image is multiplied by the given percent, then it is composited over the main image.

-**dither** apply Floyd/Steinberg error diffusion to the image
The basic strategy of dithering is to trade intensity resolution for spatial resolution by averaging the intensities of several neighboring pixels. Images which suffer from severe contouring when reducing colors can be improved with this option.

The -colors or -monochrome option is required for this option to take effect.

Use +dither to turn off dithering and to render PostScript without text or graphic aliasing.

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

Use this option to annotate an image with one or more graphic primitives. The primitives include shapes, text, transformations, and pixel operations. The shape primitives are

- **point** x,y
- **line** x0,y0 x1,y1
- **rectangle** x0,y0 x1,y1
- **roundRectangle** x0,y0 x1,y1 wc,hc
- **arc** x0,y0 x1,y1 a0,a1
- **ellipse** x0,y0 rx,ry a0,a1
- **circle** x0,y0 x1,y1
- **polyline** x0,y0 ... xn,yn
- **polygon** x0,y0 ... xn,yn
- **Bezier** x0,y0 ... xn,yn
- **path** path specification
- **image** operator x0,y0 w,h filename

The text primitive is

- **text** x0,y0 string

The transformation primitives are

- **rotate** degrees
- **translate** dx,dy
- **scale** sx,sy
- **skewX** degrees
- **skewY** degrees

The pixel operation primitives are

- **color** x0,y0 method
- **matte** x0,y0 method
The shape primitives are drawn in the color specified in the preceding -stroke option. Except for the line and point primitives, they are filled with the color specified in the preceding -fill option. For unfilled shapes, use -fill none.

Point requires a single coordinate.
Line requires a start and end coordinate.
Rectangle expects an upper left and lower right coordinate.
RoundRectangle has the upper left and lower right coordinates and the width and height of the corners.
Circle has a center coordinate and a coordinate for the outer edge.

Use Arc to circumscribe an arc within a rectangle. Arcs require a start and end point as well as the degree of rotation (e.g. 130,30 200,100 45,90).

Use Ellipse to draw a partial ellipse centered at the given point with the x-axis and y-axis radius and start and end of arc in degrees (e.g. 100,100 100,150 0,360).

Finally, polyline and polygon require three or more coordinates to define its boundaries. Coordinates are integers separated by an optional comma. For example, to define a circle centered at 100,100 that extends to 150,150 use:

```
-draw 'circle 100,100 150,150'
```

Paths (See Paths) represent an outline of an object which is defined in terms of moveto (set a new current point), lineto (draw a straight line), curveto (draw a curve using a cubic Bezier), arc (elliptical or circular arc) and closepath (close the current shape by drawing a line to the last moveto) elements. Compound paths (i.e., a path with subpaths, each consisting of a single moveto followed by one or more line or curve operations) are possible to allow effects such as “donut holes” in objects.

Use image to composite an image with another image. Follow the image keyword with the composite operator, image location, image size, and filename:

```
-draw 'image Over 100,100 225,225 image.jpg'
```

You can use 0,0 for the image size, which means to use the actual dimensions found in the image header. Otherwise, it will be scaled to the given dimensions. See -compose for a description of the composite operators.

Use text to annotate an image with text. Follow the text coordinates with a string. If the string has embedded spaces, enclose it in double quotes. Optionally you can include the image filename, type, width, height, or other image attribute by embedding special format character. See -comment for details.

For example,

```
-draw 'text 100,100 "%m:%f %wx%h"'
```
annote the image with MIFF:bird.miff 512x480 for an image titled bird.miff and whose width is 512 and height is 480.

If the first character of string is @, the text is read from a file titled by the remaining characters in the string.

**Rotate** rotates subsequent shape primitives and text primitives about the origen of the main image. If the -region option precedes the -draw option, the origen for transformations is the upper left corner of the region.

**Translate** translates them.

**Scale** scales them.

**SkewX** and **SkewY** skew them with respect to the origen of the main image or the region.

The transformations modify the current affine matrix, which is initialized from the initial affine matrix defined by the -affine option. Transformations are cumulative within the -draw option. The initial affine matrix is not affected; that matrix is only changed by the appearance of another -affine option. If another -draw option appears, the current affine matrix is reinitialized from the initial affine matrix.

Use **color** to change the color of a pixel to the fill color (see -fill). Follow the pixel coordinate with a method:

```
point
replace
floodfill
filltoborder
reset
```

Consider the target pixel as that specified by your coordinate. The **point** method recolors the target pixel. The **replace** method recolors any pixel that matches the color of the target pixel. **Floodfill** recolors any pixel that matches the color of the target pixel and is a neighbor, whereas **filltoborder** recolors any neighbor pixel that is not the border color. Finally, **reset** recolors all pixels.

Use **matte** to change the pixel matte value to transparent. Follow the pixel coordinate with a method (see the **color** primitive for a description of methods). The **point** method changes the matte value of the target pixel. The **replace** method changes the matte value of any pixel that matches the color of the target pixel. **Floodfill** changes the matte value of any pixel that matches the color of the target pixel and is a neighbor, whereas **filltoborder** changes the matte value of any neighbor pixel that is not the border color (-bordercolor). Finally **reset** changes the matte value of all pixels.

You can set the primitive color, font, and font bounding box color with -fill, -font, and -box respectively. Options are processed in command line order so be sure to use these options before the -draw option.
-edge <radius>  detect edges within an image

-emboss  emboss an image

-encoding <type>  specify the font encoding

Choose from AdobeCustom, AdobeExpert, AdobeStandard, AppleRoman, BIG5, GB2312, Latin 2, None, SJIScode, Symbol, Unicode, Wansung.

-endian <type>  specify endianness (MSB or LSB) of output image

Use +endian to revert to unspecified endianness.

-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

The color is specified using the format described in the “Color Names” section of X(1).

See -draw for further details.

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

Use this option to affect the resizing operation of an image (see -geometry). Choose from these filters:

Point
Box
Triangle
Hermite
Hanning
Hamming
Blackman
Gaussian
Quadratic
Cubic
Catrom
Mitchell
Lanczos
Bessel
Sinc
The default filter is **Lanczos**

**-flatten** flatten a sequence of images

The sequence of images is replaced by a single image created by composing each image after the first over the first image.

The sequence of images is terminated by the appearance of any option. If the **-flatten** option appears after all of the input images, all images are flattened.

**-flip** create a “mirror image”

reflect the scanlines in the vertical direction.

**-flop** create a “mirror image”

reflect the scanlines in the horizontal direction.

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

You can tag a font to specify whether it is a PostScript, TrueType, or OPTION1 font. For example, `Arial.ttf` is a TrueType font, `ps:helvetica` is PostScript, and `x:fixed` is OPTION1.

**-foreground** `<color>` define the foreground color

The color is specified using the format described in the “Color Names” section of `X(1)`.

**-format** `<type>` the image format type

This option will convert any image to the image format you specify. See `ImageMagick(1)` for a list of image format types supported by `ImageMagick`.

By default the file is written to its original name. However, if the filename extension matches a supported format, the extension is replaced with the image format type specified with **-format**. For example, if you specify `tiff` as the format type and the input image filename is `image.gif`, the output image filename becomes `image.tiff`.

**-format** `<string>` output formatted image characteristics

Use this option to print information about the image in a format of your choosing. You can include the image filename, type, width, height, or other image attributes by embedding special format characters:
%b  file size
%c  comment
%d  directory
%e  filename extension
%f  filename
%h  height
%i  input filename
%k  number of unique colors
%l  label
%m  magick
%n  number of scenes
%o  output filename
%p  page number
%q  quantum depth
%s  scene number
%t  top of filename
%u  unique temporary filename
%w  width
%x  x resolution
%y  y resolution
%#  signature
\n  newline
\r  carriage return

For example,

    -format "%m:%f %wx%h"

displays MIFF:bird.miff 512x480 for an image titled bird.miff and whose width is 512 and height is 480.

If the first character of string is @, the format is read from a file titled by the remaining characters in the string.

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

See -geometry for details about the geometry specification. The -frame option is not affected by the -gravity option.

The color of the border is specified with the -mattecolor command line option.

-frame  include the X window frame in the imported image

-fuzz <distance>{ %}  colors within this distance are considered equal
A number of algorithms search for a target color. By default the color must be exact. Use this option to match colors that are close to the target color in RGB space. For example, if you want to automatically trim the edges of an image with `-trim` but the image was scanned and the target background color may differ by a small amount. This option can account for these differences.

The *distance* can be in absolute intensity units or, by appending “%”, as a percentage of the maximum possible intensity (255 or 65535).

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

The same color image displayed on two different workstations may look different due to differences in the display monitor. Use gamma correction to adjust for this color difference. Reasonable values extend from 0.8 to 2.3.

You can apply separate gamma values to the red, green, and blue channels of the image with a gamma value list delineated with slashes (e.g., 1.7/2.3/1.2).

Use `+gamma value` to set the image gamma level without actually adjusting the image pixels. This option is useful if the image is of a known gamma but not set as an image attribute (e.g. PNG images).

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

Use the given radius and standard deviation (sigma).

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

By default, the window size is the image size and the location is chosen by you when it is mapped.

By default, the width and height are maximum values. That is, the image is expanded or contracted to fit the width and height value while maintaining the aspect ratio of the image. *Append an exclamation point to the geometry to force the image size to exactly the size you specify.* For example, if you specify 640x480! the image width is set to 640 pixels and height to 480.

If only the width is specified, the width assumes the value and the height is chosen to maintain the aspect ratio of the image. Similarly, if only the height is specified (e.g., `-geometry x256`), the width is chosen to maintain the aspect ratio.

To specify a percentage width or height instead, append %. The image size is multiplied by the width and height percentages to obtain the final image dimensions. To increase the size of an image, use a value greater than 100 (e.g. 125%). To decrease an image’s size, use a percentage less than 100.

Use @ to specify the maximum area in pixels of an image.
Use > to change the dimensions of the image only if its width or height exceeds the geometry specification. < resizes the image only if both of its dimensions are less than the geometry specification. For example, if you specify ‘640x480>’ and the image size is 256x256, the image size does not change. However, if the image is 512x512 or 1024x1024, it is resized to 480x480. Enclose the geometry specification in quotation marks to prevent the < or > from being interpreted by your shell as a file redirection.

When used with animate and display, offsets are handled in the same manner as in X(1) and the -gravity option is not used. If the x is negative, the offset is measured leftward from the right edge of the screen to the right edge of the image being displayed. Similarly, negative y is measured between the bottom edges. The offsets are not affected by “%”; they are always measured in pixels.

When used as a composite option, -geometry gives the dimensions of the image and its location with respect to the composite image. If the -gravity option is present with NorthEast, East, or SouthEast gravity, the x represents the distance from the right edge of the image to the right edge of the composite image. Similarly, if the -gravity option is present with SouthWest, South, or SouthEast gravity, y is measured between the bottom edges. Accordingly, a positive offset will never point in the direction outside of the image. The offsets are not affected by “%”; they are always measured in pixels. To specify the dimensions of the composite image, use the -resize option.

When used as a convert, import or mogrify option, -geometry is synonymous with -resize and specifies the size of the output image. The offsets, if present, are ignored.

When used as a montage option, -geometry specifies the image size and border size for each tile; default is 256x256+0+0. Negative offsets (border dimensions) are meaningless. The -gravity option affects the placement of the image within the tile; the default gravity for this purpose is Center. If the “%” sign appears in the geometry specification, the tile size is the specified percentage of the original dimensions of the first tile. To specify the dimensions of the montage, use the -resize option.

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

Choices are: NorthWest, North, NorthEast, West, Center, East, SouthWest, South, SouthEast.

The direction you choose specifies where to position the text or other graphic primitive when annotating the image. For example Center gravity forces the text to be centered within the image. By default, the image gravity is NorthWest. See -draw for more details about graphic primitives.

The -gravity option is also used in concert with the -geometry option and other options that take <geometry> as a parameter, such as the -crop option. See
-geometry for details of how the -gravity option interacts with the <x> and <y> parameters of a geometry specification.

When used as an option to composite, -gravity gives the direction that the image gravitates within the composite.

When used as an option to montage, -gravity gives the direction that an image gravitates within a tile. The default gravity is Center for this purpose.

-help print usage instructions

-iconGeometry <geometry> specify the icon geometry

Offsets, if present in the geometry specification, are handled in the same manner as the -geometry option, using X11 style to handle negative offsets.

-iconic iconic animation

-immutable make image immutable

-implode <factor> implode image pixels about the center

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

Use this option to affect the the color management operation of an image (see -profile). Choose from these intents: Absolute, Perceptual, Relative, Saturati0n

The default intent is undefined.

-interlace <type> the type of interlacing scheme

Choices are: None, Line, Plane, or Partition. The default is None.

This option is used to specify the type of interlacing scheme for raw image formats such as RGB or YUV.

None means do not interlace (RGBRGBRGBRGBRGBRGB...),

Line uses scanline interlacing (RRR...GGG...BBB...RRR...GGG...BBB...), and

Plane uses plane interlacing (RRRRRR...GGGGGG...BBBBBB...).

Partition is like plane except the different planes are saved to individual files (e.g. image.R, image.G, and image.B).

Use Line or Plane to create an interlaced PNG or GIF or progressive JPEG image.
-label <name> assign a label to an image

Use this option to assign a specific label to the image. Optionally you can include
the image filename, type, width, height, or other image attribute by embedding
special format character. See -comment for details.

For example,

    -label "%m:%f %wx%h"

produces an image label of MIFF:bird.miff 512x480 for an image titled bird.miff
and whose width is 512 and height is 480.

If the first character of string is @, the image label is read from a file titled by
the remaining characters in the string.

When converting to PostScript, use this option to specify a header string to print
above the image. Specify the label font with -font.

-level <value> adjust the level of image contrast

Give three point values delineated with commas: black, mid, and white (e.g.
10,1.0,65000). The white and black points range from 0 to MaxRGB and mid
ranges from 0 to 10.

-linewidth the line width for subsequent draw operations

-list <type> the type of list

Choices are: Delegate, Format, Magic, Module, or Type.

This option lists entries from the ImageMagick configuration files.

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

A value other than zero forces the animation to repeat itself up to iterations
times.

-magnify <factor> magnify the image

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

[convert or mogrify]

By default, color reduction chooses an optimal set of colors that best represent
the original image. Alternatively, you can choose a particular set of colors from
an image file with this option.
Use `+map` to reduce all images in the image sequence that follows to a single optimal set of colors that best represent all the images. The sequence of images is terminated by the appearance of any option. If the `+map` option appears after all of the input images, all images are mapped.

`-map <type>` display image using this type.

[animate or display]

Choose from these *Standard Colormap* types:

```
best
default
gray
red
green
blue
```

The *X server* must support the *Standard Colormap* you choose, otherwise an error occurs. Use list as the type and display searches the list of colormap types in *top-to-bottom* order until one is located. See *xstdcmap(1)* for one way of creating Standard Colormaps.

`-mask <filename>` Specify a clipping mask

The image read from the file is used as a clipping mask. It must have the same dimensions as the image being masked.

If the mask image contains an opacity channel, the opacity of each pixel is used to define the mask. Otherwise, the intensity (gray level) of each pixel is used.

Use `+mask` to remove the clipping mask.

It is not necessary to use `-clip` to activate the mask; `-clip` is implied by `-mask`.

`-matte` store matte channel if the image has one

If the image does not have a matte channel, create an opaque one.

Use `+matte` to ignore the matte channel and to avoid writing a matte channel in the output file.

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

The color is specified using the format described in the “Color Names” section of *X(1)*.

`-median <radius>` apply a median filter to the image
-mode <value>  mode of operation

-modulate <value>  vary the brightness, saturation, and hue of an image
   Specify the percent change in brightness, the color saturation, and the hue separated by commas. For example, to increase the color brightness by 20% and decrease the color saturation by 10% and leave the hue unchanged, use: -modulate 120,90.

-monochrome  transform the image to black and white

-morph <frames>  morphs an image sequence
   Both the image pixels and size are linearly interpolated to give the appearance of a meta-morphosis from one image to the next.
   The sequence of images is terminated by the appearance of any option. If the -morph option appears after all of the input images, all images are morphed.

-mosaic  create a mosaic from an image sequence
   The -page option is used to locate the images within the mosaic.
   The sequence of images is terminated by the appearance of any option. If the -mosaic option appears after all of the input images, all images are included in the mosaic.

-name  name an image

-negate  replace every pixel with its complementary color
   The red, green, and blue intensities of an image are negated. White becomes black, yellow becomes blue, etc. Use +negate to only negate the grayscale pixels of the image.

-noise <radius|type>  add or reduce noise in an image
   The principal function of noise peak elimination filter is to smooth the objects within an image without losing edge information and without creating undesired structures. The central idea of the algorithm is to replace a pixel with its next neighbor in value within a pixel window, if this pixel has been found to be noise. A pixel is defined as noise if and only if this pixel is a maximum or minimum within the pixel window.
   Use radius to specify the width of the neighborhood.
Use \texttt{+noise} followed by a noise type to add noise to an image. Choose from these noise types:

\begin{itemize}
\item Uniform
\item Gaussian
\item Multiplicative
\item Impulse
\item Laplacian
\item Poisson
\end{itemize}

\texttt{-noop} NOOP (no option)

The \texttt{-noop} option can be used to terminate a group of images and reset all options to their default values, when no other option is desired.

\texttt{-normalize} transform image to span the full range of color values

This is a contrast enhancement technique.

\texttt{-opaque <color>} change this color to the pen color within the image

The color is specified using the format described in the “Color Names” section of \texttt{x(1)}.

See \texttt{-fill} for more details.

\texttt{-page <width>x<height>{+-}{x}{+-}{y}{%}{l}{r}{t}{b}} size and location of an image canvas

Use this option to specify the dimensions of the \texttt{PostScript} page in dots per inch or a \texttt{TEXT} page in pixels. The choices for a \texttt{PostScript} page are:

\begin{verbatim}
11x17  792  1224
Ledger 1224  792
Legal  612  1008
Letter  612   792
LetterSmall  612   792
ArchE  2592  3456
ArchD  1728  2592
ArchC  1296  1728
ArchB   864  1296
ArchA   648   864
A0     2380  3368
A1     1684  2380
A2     1190  1684
\end{verbatim}
For convenience you can specify the page size by media (e.g. A4, Ledger, etc.). Otherwise, `-page` behaves much like `-geometry` (e.g. `-page letter+43+43`).

To position a GIF image, use `-page {+-}<x>{+-}<y>` (e.g. `-page +100+200`).

For a PostScript page, the image is sized as in `-geometry` and positioned relative to the lower left hand corner of the page by `{+-}<x offset>{+-}<y offset>`. Use `-page 612x792`, for example, to center the image within the page. If the image size exceeds the PostScript page, it is reduced to fit the page. The default gravity for the `-page` option is `NorthWest`, i.e., positive `x` and `y` offset are measured rightward and downward from the top left corner of the page, unless the `-gravity` option is present with a value other than `NorthWest`.

The default page dimensions for a TEXT image is 612x792.

This option is used in concert with `-density`.

```
-paint <radius> simulate an oil painting

Each pixel is replaced by the most frequent color in a circular neighborhood whose width is specified with `radius`.
```
-pause <seconds>  pause between animation loops [animate]
    Pause for the specified number of seconds before repeating the animation.

-pause <seconds>  pause between snapshots [import]
    Pause for the specified number of seconds before taking the next snapshot.

-pen <color>  specify the pen color for drawing operations
    The color is specified using the format described in the “Color Names” section
    of X(1).
    This option is deprecated; use -fill instead.

-ping  efficiently determine image characteristics

-pointsize <value>  pointsize of the PostScript, OPTION1, or TrueType font

-preview <type>  image preview type
    Use this option to affect the preview operation of an image (e.g. convert
    -preview Gamma Preview:gamma.png). Choose from these previews:
    Rotate
    Shear
    Roll
    Hue
    Saturation
    Brightness
    Gamma
    Spiff
    Dull
    Grayscale
    Quantize
    Despeckle
    ReduceNoise
    Add Noise
    Sharpen
    Blur
    Threshold
    EdgeDetect
    Spread
    Shade
    Raise
ImageMagick Utilities

Segment
Solarize
Swirl
Implode
Wave
OilPaint
CharcoalDrawing
JPEG

The default preview is JPEG.

-process <command> process a sequence of images

The sequence of images is terminated by the appearance of any option. If the -process option appears after all of the input images, all images are processed.

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

-profile filename adds an ICM (ICC color management), IPTC (newswire information), or a generic profile to the image.

Use +profile icm, +profile iptc, or +profile profile_name to remove the respective profile. Use identify -verbose to find out what profiles are in the image file. Use +profile "*" to remove all profiles.

To extract a profile, the -profile option is not used. Instead, simply write the file to an image format such as APP1, 8BIM, ICM, or IPTC.

For example, to extract the Exif data (which is stored in JPEG files in the APP1 profile), use

    convert cockatoo.jpg exifdata.app1

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

For the JPEG and MPEG image formats, quality is 0 (lowest image quality and highest compression) to 100 (best quality but least effective compression). The default quality is 75. Use the -sampling_factor option to specify the factors for chroma downsampling.

For the MIFF image format, quality/10 is the zlib compression level, which is 0 (worst but fastest compression) to 9 (best but slowest). It has no effect on the image appearance, since the compression is always lossless.

For the MNG and PNG image formats, the quality value sets the zlib compression level (quality / 10) and filter-type (quality % 10). Compression levels range from 0 (fastest compression) to 100 (best but slowest). For compression level 0,
the Huffman-only strategy is used, which is fastest but not necessarily the worst compression.

If filter-type is 4 or less, the specified filter-type is used for all scanlines:

0: none
1: sub
2: up
3: average
4: Paeth

If filter-type is 5, adaptive filtering is used when quality is greater than 50 and the image does not have a color map, otherwise no filtering is used.

If filter-type is 6, adaptive filtering with minimum-sum-of-absolute-values is used.

Only if the output is MNG, if filter-type is 7, the LOCO color transformation and adaptive filtering with minimum-sum-of-absolute-values are used.

The default is quality is 75, which means nearly the best compression with adaptive filtering. The quality setting has no effect on the appearance of PNG and MNG images, since the compression is always lossless.

For further information, see the PNG specification.

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

This will create a 3-D effect. See -geometry for details about the geometry specification. Offsets are not used.

Use -raise to create a raised effect, otherwise use +raise.

-region <width>x<height>{+}x{+}y apply options to a portion of the image

The x and y offsets are treated in the same manner as in -crop.

-remote perform a remote operation

The only command recognized at this time is the name of an image file to load.

-resize <width>x<height>%[@]{!}{<}{>} resize an image

This is an alias for the -geometry option and it behaves in the same manner. If the -filter option precedes the -resize option, the specified filter is used.

There are some exceptions:
When used as a composite option, -resize conveys the preferred size of the output image, while -geometry conveys the size and placement of the composite image within the main image.

When used as a montage option, -resize conveys the preferred size of the montage, while -geometry conveys information about the tiles.

-rolls {+-} <x> {+-} <y> roll an image vertically or horizontally

See -geometry for details the geometry specification. The x and y offsets are not affected by the -gravity option.

A negative x offset rolls the image left-to-right. A negative y offset rolls the image top-to-bottom.

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

Use > to rotate the image only if its width exceeds the height. < rotates the image only if its width is less than the height. For example, if you specify -rotate "-90" and the image size is 480x640, the image is not rotated. However, if the image is 640x480, it is rotated by -90 degrees. If you use > or <, enclose it in quotation marks to prevent it from being misinterpreted as a file redirection.

Empty triangles left over from rotating the image are filled with the color defined as background (class backgroundColor). See X(1) for details.

-sample <geometry> scale image with pixel sampling

See -geometry for details about the geometry specification. -sample ignores the -filter selection if the -filter option is present. Offsets, if present in the geometry string, are ignored, and the -gravity option has no effect.

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

This option specifies the sampling factors to be used by the JPEG encoder for chroma downsampling. If this option is omitted, the JPEG library will use its own default values. When reading or writing the YUV format and when writing the M2V (MPEG-2) format, use -sampling_factor 2x1 to specify the 4:2:2 downsampling method.

-scale <geometry> scale the image.

See -geometry for details about the geometry specification. -scale uses a simpler, faster algorithm, and it ignores the -filter selection if the -filter option is present. Offsets, if present in the geometry string, are ignored, and the -gravity option has no effect.
-scene <value>  set scene number

This option sets the scene number of an image or the first image in an image sequence.

-scenes <value-value>  range of image scene numbers to read

Each image in the range is read with the filename followed by a period (.) and
the decimal scene number. You can change this behavior by embedding a %d,
%0Nd, %o, %0No, %x, or %0Nx printf format specification in the file name.
For example,

montage -scenes 5-7 image.miff

makes a montage of files image.miff.5, image.miff.6, and image.miff.7, and

animate -scenes 0-12 image%02d.miff

animates files image00.miff, image01.miff, through image12.miff.

-screen  specify the screen to capture

This option indicates that the GetImage request used to obtain the image should
be done on the root window, rather than directly on the specified window. In this
way, you can obtain pieces of other windows that overlap the specified window,
and more importantly, you can capture menus or other popups that are independ-
ent windows but appear over the specified window.

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

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

Segment an image by analyzing the histograms of the color components and
identifying units that are homogeneous with the fuzzy c-means technique.

Specify cluster threshold as the number of pixels in each cluster must exceed
the the cluster threshold to be considered valid. Smoothing threshold eliminates
noise in the second derivative of the histogram. As the value is increased, you
can expect a smoother second derivative. The default is 1.5. See “Image Seg-
mentation” in the manual page for display for details.

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

Specify azimuth and elevation as the position of the light source. Use +shade to
return the shading results as a grayscale image.
-**shadow** `<radius>x<sigma>`  shadow the montage

-**shared_memory**  use shared memory
This option specifies whether the utility should attempt use shared memory for pixmaps. ImageMagick must be compiled with shared memory support, and the display must support the MIT-SHM extension. Otherwise, this option is ignored. The default is True.

-**sharpen** `<radius>x<sigma>`  sharpen the image
Use a Gaussian operator of the given radius and standard deviation (sigma).

-**shave** `<width>x<height>`  shave pixels from the image edges
Specify the width of the region to be removed from both sides of the image and the height of the regions to be removed from top and bottom.

-**shear** `<x degrees>x<y degrees>`  shear the image along the X or Y axis
Use the specified positive or negative shear angle.
Shearing slides one edge of an image along the X or Y axis, creating a parallelogram. An X direction shear slides an edge along the X axis, while a Y direction shear slides an edge along the Y axis. The amount of the shear is controlled by a shear angle. For X direction shears, `x degrees` is measured relative to the Y axis, and similarly, for Y direction shears `y degrees` is measured relative to the X axis.
Empty triangles left over from shearing the image are filled with the color defined as background (class backgroundColor). See X(1) for details.

-**silent**  operate silently

-**size** `<width>x<height>{+offset}`  width and height of the image
Use this option to specify the width and height of raw images whose dimensions are unknown such as GRAY, RGB, or CMYK. In addition to width and height, use -size with an offset to skip any header information in the image or tell the number of colors in a MAP image file, (e.g. -size 640x512+256).
For Photo CD images, choose from these sizes:

- 192x128
- 384x256
- 768x512
- 1536x1024
- 3072x2048
Finally, use this option to choose a particular resolution layer of a JBIG or JPEG image (e.g. -size 1024x768).

**-snaps <value>**  number of screen snapshots

Use this option to grab more than one image from the X server screen, to create an animation sequence.

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

Specify factor as the percent threshold of the intensity (0 - 99.9%).

This option produces a solarization effect seen when exposing a photographic film to light during the development process.

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

Amount defines the size of the neighborhood around each pixel to choose a candidate pixel to swap.

**-stegano <offset>**  hide watermark within an image

Use an offset to start the image hiding some number of pixels from the beginning of the image. Note this offset and the image size. You will need this information to recover the steganographic image (e.g. display -size 320x256+35 stegano:image.png).

**-stereo**  composite two images to create a stereo anaglyph

The left side of the stereo pair is saved as the red channel of the output image. The right side is saved as the green channel. Red-green stereo glasses are required to properly view the stereo image.

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

The color is specified using the format described in the “Color Names” section of X(1).

See -draw for further details.

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

See -draw for further details.

**-swirl <degrees>**  swirl image pixels about the center

Degrees defines the tightness of the swirl.
-text_font <name>  font for writing fixed-width text

Specifies the name of the preferred font to use in fixed (typewriter style) formatted text. The default is 14 point Courier.

You can tag a font to specify whether it is a PostScript, TrueType, or OPTION1 font. For example, Courier.ttf is a TrueType font and x:fixed is OPTION1.

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

-threshold <value>  threshold the image

Create a bi-level image such that any pixel intensity that is equal or exceeds the threshold is reassigned the maximum intensity otherwise the minimum intensity.

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

-tile <geometry>  layout of images [montage]

-title <string>  assign title to displayed image [animate, display, montage]

Use this option to assign a specific title to the image. This is assigned to the image window and is typically displayed in the window title bar. Optionally you can include the image filename, type, width, height, or other image attribute by embedding special format characters:

- %b file size
- %c comment
- %d directory
- %e filename extension
- %f filename
- %h height
- %i input filename
- %k number of unique colors
- %l label
- %m magick
- %n number of scenes
- %o output filename
- %p page number
- %q quantum depth
- %s scene number
- %t top of filename
- %u unique temporary filename
For example,

```
-title "%m:%f %wx%h"
```

produces an image title of MIFF:bird.miff 512x480 for an image titled bird.miff and whose width is 512 and height is 480.

**-transform** transform the image

This option applies the transformation matrix from a previous **-affine** option.

```
convert -affine 2,2,-2,2,0,0 -transform bird.ppm bird.jpg
```

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

The color is specified using the format described in the “Color Names” section of X(1).

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

Normally, this integer value is zero or one. A zero or one tells display to choose an optimal tree depth for the color reduction algorithm.

An optimal depth generally allows the best representation of the source image with the fastest computational speed and the least amount of memory. However, the default depth is inappropriate for some images. To assure the best representation, try values between 2 and 8 for this parameter. Refer to quantize for more details.

The **-colors** or **-monochrome** option is required for this option to take effect.

**-trim** trim an image

This option removes any edges that are exactly the same color as the corner pixels. Use **-fuzz** to make **-trim** remove edges that are nearly the same color as the corner pixels.
-**type** <type> the image type

Choose from: **Bilevel, Grayscale, Palette, PaletteMatte, TrueColor, TrueColorMatte, ColorSeparation, ColorSeparationMatte**, or **Optimize**.

Normally, when a format supports different subformats such as grayscale and truecolor, the encoder will try to choose an efficient subformat. The **-type** option can be used to override this behavior. For example, to prevent a JPEG from being written in grayscale format even though only gray pixels are present, use

```
convert bird.pgm -type TrueColor bird.jpg
```

Similarly, using **-type TrueColorMatte** will force the encoder to write an alpha channel even though the image is opaque, if the output format supports transparency.

-**update** <seconds> detect when image file is modified and redisplay.

Suppose that while you are displaying an image the file that is currently displayed is over-written. **display** will automatically detect that the input file has been changed and update the displayed image accordingly.

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

Choose from: **Undefined, PixelsPerInch**, or **PixelsPerCentimeter**.

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

Use the given radius and standard deviation (sigma).

-**use_pixmap** use the pixmap

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

This information is printed: image scene number; image name; image size; the image class (**DirectClass** or **PseudoClass**); the total number of unique colors; and the number of seconds to read and transform the image. Refer to miff for a description of the image class.

If **-colors** is also specified, the total unique colors in the image and color reduction error values are printed. Refer to quantize for a description of these values.

-**view** <string> FlashPix viewing parameters
-visual <type>  animate images using this X visual type

Choose from these visual classes:

  StaticGray
  GrayScale
  StaticColor
  PseudoColor
  TrueColor
  DirectColor
  default
  visual id

The X server must support the visual you choose, otherwise an error occurs. If a visual is not specified, the visual class that can display the most simultaneous colors on the default screen is chosen.

-watermark <brightness>x<saturation>  percent brightness and saturation of a watermark

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

Specify amplitude and wavelength of the wave.

-window <id>  make image the background of a window

id can be a window id or name. Specify root to select X’s root window as the target window.

By default the image is tiled onto the background of the target window. If backdrop or -geometry are specified, the image is surrounded by the background color. Refer to X RESOURCES for details.

The image will not display on the root window if the image has more unique colors than the target window colormap allows. Use -colors to reduce the number of colors.

-window_group  specify the window group

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

The image sequence following the -write filename option is written out, and then processing continues with the same image in its current state if there are additional options. To restore the image to its original state after writing it, use the +write filename option.
**-write <filename>**  write the image to a file [display]

If filename already exists, you will be prompted as to whether it should be over-written.

By default, the image is written in the format that it was read in as. To specify a particular image format, prefix filename with the image type and a colon (e.g., ps:image) or specify the image type as the filename suffix (e.g., image.ps). See convert(1) for a list of valid image formats. Specify file as - for standard output. If file has the extension .Z or .gz, the file size is compressed using compress or gzip respectively. Precede the image file name with | to pipe to a system command.

Use -compress to specify the type of image compression.

The equivalent X resource for this option is writeFilename (class WriteFilename). See "X Resources" in the manual page for display for details.

### 1.5 Files and Formats

By default, the image format is determined by its magic number, i.e., the first few bytes of the file. To specify a particular image format, precede the filename with an image format name and a colon (i.e., ps:image) or specify the image type as the filename suffix. The magic number takes precedence over the filename suffix and the prefix takes precedence over the magic number and the suffix in input files. The prefix takes precedence over the filename suffix in output files. To read the "built-in" formats (GRANITE, H, LOGO, NETSCAPE, PLASMA, and ROSE) use a prefix (including the colon) without a filename or suffix. To read the XC format, follow the colon with a color specification. To read the CAPTION format, follow the colon with a text string or with a filename prefixed with the at symbol (@).

When you specify X as your image type, the filename has special meaning. It specifies an X window by id, name, or root. If no filename is specified, the window is selected by clicking the mouse in the desired window.

Specify input_file as - for standard input, output_file as - for standard output. If input_file has the extension .Z or .gz, the file is uncompressed with uncompress or gunzip respectively. If output_file has the extension .Z or .gz, the file is compressed using with compress or gzip respectively.

Finally, when running on platforms that allow it, precede the image file name with | to pipe to or from a system command (this feature is not available on VMS, Win32 and Macintosh platforms).

Use an optional index enclosed in brackets after an input file name to specify a desired subimage of a multi-resolution image format like Photo CD (e.g. img0001.pcd[4]) or a range for MPEG images (e.g. video.mpg[50-75]). A subimage specification can be disjoint (e.g. image.tiff[2,7,4]). For
raw images, specify a subimage with a geometry (e.g. \texttt{\-size 640x512 image.rgb[320x256+50+50]}). Single images are written with the filename you specify. However, multi-part images (e.g., a multi-page PostScript document with \texttt{+adjoin} specified) are written with the filename followed by a period (\texttt{.}) and the scene number. You can change this behavior by embedding a \texttt{\%d}, \texttt{\%0Nd}, \texttt{\%o}, \texttt{\%0No}, \texttt{\%x}, or \texttt{\%0Nx} \texttt{printf} format specification in the file name. For example,

\texttt{image\%02d.miff}

writes files \texttt{image00.miff}, \texttt{image01.miff}, etc.

When running a commandline utility, you can prepend an at sign \texttt{@} to a filename to read a list of image filenames from that file. This is convenient in the event you have too many image filenames to fit on the command line.

1.6 Environment

\texttt{DISPLAY} To get the default host, display number, and screen.

1.7 Authors

\textit{John Cristy, ImageMagick Studio LLC,}
\textit{Glenn Randers-Pehrson, ImageMagick Studio LLC.}
2 Animate

2.1 NAME

animate - animate a sequence of images

2.2 Synopsis

animate [ options ...] file [ [ options ...] file ...]

2.3 Description

Animate displays a sequence of images on any workstation display running an X server. animate first determines the hardware capabilities of the workstation. If the number of unique colors in an image is less than or equal to the number the workstation can support, the image is displayed in an X window. Otherwise the number of colors in the image is first reduced to match the color resolution of the workstation before it is displayed.

This means that a continuous-tone 24 bits-per-pixel image can display on a 8 bit pseudo-color device or monochrome device. In most instances the reduced color image closely resembles the original. Alternatively, a monochrome or pseudo-color image sequence can display on a continuous-tone 24 bits-per-pixel device.

To help prevent color flashing on X server visuals that have colormaps, animate creates a single colormap from the image sequence. This can be rather time consuming. You can speed this operation up by reducing the colors in the image before you “animate” them. Use mogrify to color reduce the images to a single colormap. See mogrify(1) for details. Alternatively, you can use a Standard Colormap; or a static, direct, or true color visual. You can define a Standard
Colormap with \textit{xstdcmap}. See \textit{xstdcmap(1)} for details. This method is recommended for colormapped X server because it eliminates the need to compute a global colormap.

### 2.4 Examples

To animate a set of images of a cockatoo, use:

\begin{verbatim}
animate cockatoo.*
\end{verbatim}

To animate a cockatoo image sequence while using the Standard Colormap \textit{best}, use:

\begin{verbatim}
xstdcmap -best
animate -map best cockatoo.*
\end{verbatim}

To animate an image of a cockatoo without a border centered on a backdrop, use:

\begin{verbatim}
animate +borderwidth -backdrop cockatoo.*
\end{verbatim}

### 2.5 Options

For a more detailed description of each option, see \textit{ImageMagick(1)}.

\textbf{-backdrop <color>} display the image centered on a backdrop.

\textbf{-background <color>} the background color

\textbf{-bordercolor <color>} the border color

\textbf{-borderwidth <geometry>} the border width

\textbf{-cache <threshold>} megabytes of memory available to the pixel cache

\textbf{-chop <width>x<height>{+}-}<x>{+/-}<y>{%} remove pixels from the interior of an image
-colormap <type>  define the colormap type

-colors <value>  preferred number of colors in the image

-colorspace <value>  the type of colorspace

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

-debug  enable debug printout

-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

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

-dispose <method>  GIF disposal method

-dither  apply Floyd/Steinberg error diffusion to the image

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

-foreground <color>  define the foreground color

-gamma <value>  level of gamma correction

-geometry <width>x<height>{+-}<x>{+-}<y>{%}@>{f}{l}<c><g>  preferred size and location of the Image window.

-help  print usage instructions
-iconGeometry <geometry> specify the icon geometry

-iconic iconic animation

-interlace <type> the type of interlacing scheme

-map <type> display image using this type.

-matte store matte channel if the image has one

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

-monochrome transform the image to black and white

-name name an image

-noop NOOP (no option)

-pause <seconds> pause between animation loops [animate]

-remote perform a remote operation

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

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

-scenes <value-value> range of image scene numbers to read

-shared_memory use shared memory

-size <width>x<height>{+offset} width and height of the image
-text_font <name>   font for writing fixed-width text

-title <string>    assign title to displayed image [animate, display, montage]

-treedepth <value>  tree depth for the color reduction algorithm

-verbose    print detailed information about the image

-visual <type>    animate images using this X visual type

-window <id>    make image the background of a window

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

Any option you specify on the command line remains in effect for the group of images following it, until the group is terminated by the appearance of any option or -noop. For example, to animate three images, the first with 32 colors, the second with an unlimited number of colors, and the third with only 16 colors, use:

animate -colors 32 cockatoo.1 -noop cockatoo.2
         -colors 16 cockatoo.3

Animate options can appear on the command line or in your X resources file. See X(1). Options on the command line supersede values specified in your X resources file. Image filenames may appear in any order on the command line if the image format is MIFF (refer to miff(5) and the scene keyword is specified in the image. Otherwise the images will display in the order they appear on the command line.

2.6 Mouse Buttons

Press any button to map or unmap the Command widget. See the next section for more information about the Command widget.
2.7 Command Widget

The Command widget lists a number of sub-menus and commands. They are

**Animate**
- Open
- Play
- Step
- Repeat
- Auto Reverse

**Speed**
- Faster
- Slower

**Direction**
- Forward
- Reverse

**Image Info**

**Help**

**Quit**

Menu items with an indented triangle have a sub-menu. They are represented above as the indented items. To access a sub-menu item, move the pointer to the appropriate menu and press a button and drag. When you find the desired sub-menu item, release the button and the command is executed. Move the pointer away from the sub-menu if you decide not to execute a particular command.

2.8 Keyboard Accelerators

**Ctlo+O** Press to load an image from a file.

**space** Press to display the next image in the sequence.

**<** Press to speed-up the display of the images. Refer to `-delay` for more information.

**>** Press to slow the display of the images. Refer to `-delay` for more information.

**?** Press to display information about the image. Press any key or button to erase the information. This information is printed: image name; image size; and the total number of unique colors in the image.

**F1** Press to display helpful information about `animate(1)`.

**Ctlo-q** Press to discard all images and exit program.

2.9 X Resources

**Animate** options can appear on the command line or in your X resource file. Options on the command line supersede values specified in your X resource file. See X(1) for more information on X resources.

All **animate** options have a corresponding X resource. In addition, the **animate** program uses the following X resources:
**background** (*class Background*)

Specifies the preferred color to use for the Image window background. The default is #ccc.

**borderColor** (*class BorderColor*)

Specifies the preferred color to use for the Image window border. The default is #ccc.

**borderWidth** (*class BorderWidth*)

Specifies the width in pixels of the Image window border. The default is 2.

**font** (*class Font or FontList*)

Specifies the name of the preferred font to use in normal formatted text. The default is 14 point Helvetica.

**foreground** (*class Foreground*)

Specifies the preferred color to use for text within the Image window. The default is black.

**geometry** (*class geometry*)

Specifies the preferred size and position of the image window. It is not necessarily obeyed by all window managers. Offsets, if present, are handled in X(1) style. A negative x offset is measured from the right edge of the screen to the right edge of the icon, and a negative y offset is measured from the bottom edge of the screen to the bottom edge of the icon.

**iconGeometry** (*class IconGeometry*)

Specifies the preferred size and position of the application when iconified. It is not necessarily obeyed by all window managers. Offsets, if present, are handled in the same manner as in class Geometry.

**iconic** (*class Iconic*)

This resource indicates that you would prefer that the application’s windows initially not be visible as if the windows had be immediately iconified by you. Window managers may choose not to honor the application’s request.

**matteColor** (*class MatteColor*)

Specify the color of windows. It is used for the backgrounds of windows, menus, and notices. A 3D effect is achieved by using highlight and shadow colors derived from this color. Default value: #ddd.

**name** (*class Name*)

This resource specifies the name under which resources for the application should be found. This resource is useful in shell aliases to distinguish between invocations of an application, without resorting to creating links to alter the executable file name. The default is the application name.

**sharedMemory** (*class SharedMemory*)

This resource specifies whether animate should attempt use shared memory for pixmaps. ImageMagick must be compiled with shared memory support, and the display must support the MIT-SHM extension. Otherwise, this resource is ignored. The default is True.

**text_font** (*class textFont*)

Specifies the name of the preferred font to use in fixed (typewriter style) formatted text. The default is 14 point Courier.
title (*class* Title)

This resource specifies the title to be used for the Image window. This information is sometimes used by a window manager to provide some sort of header identifying the window. The default is the image file name.

2.10 Environment

*DISPLAY* To get the default host, display number, and screen.

2.11 Acknowledgements

The MIT X Consortium for making network transparent graphics a reality.

*Michael Halle*, Spatial Imaging Group at MIT, for the initial implementation of Alan Paeth’s image rotation algorithm.

*David Pensak*, duPont, for providing a computing environment that made this program possible.

*Paul Raveling*, USC Information Sciences Institute. The spatial subdivision color reduction algorithm is based on his Img software.

2.12 Authors

*John Cristy*, ImageMagick Studio LLC,

*Glenn Randers-Pehrson*, ImageMagick Studio LLC.
3 Composite

3.1 NAME

composite - composite images together.

3.2 Synopsis

```
composite [ options ... ] image composite [ mask ] composited
```

3.3 Description

composite composite images to create new images. composite is the base image and image contains the changes. composited is the result, and normally has the same dimensions as composite.

The optional mask can be used to provide matte information for composite when it has none or if you want a different mask. A mask image is typically grayscale and the same size as composite. If mask is not grayscale, it is converted to grayscale and the resulting intensities are used as matte information.

3.4 Examples

To composite an image of a cockatoo with a perch, use:

```
composite cockatoo.miff perch.ras composite.miff
```

To compute the difference between images in a series, use:

```
```
composite -compose difference series.2 series.1
difference.miff

To composite an image of a cockatoo with a perch starting at location (100,150), use:

composite -geometry +100+150 cockatoo.miff
       perch.ras composite.miff

To tile a logo across your image of a cockatoo, use

convert +shade 30x60 cockatoo.miff mask.miff
composite -compose bumpmap -tile logo.png
       cockatoo.miff mask.miff composite.miff

To composite a red, green, and blue color plane into a single composite image, try

composite -compose CopyGreen green.png red.png
       red-green.png
composite -compose CopyBlue blue.png red-green.png
       composite.png

3.5 Options

Options are processed in command line order. Any option you specify on the command line remains in effect only for the image that follows. All options are reset to their default values after each image is read.

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

-**background** <color>  the background color

-**cache** <threshold>  megabytes of memory available to the pixel cache

-**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

-**debug**  enable debug printout

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

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

-**displace** <horizontal scale>x<vertical scale>  shift image pixels as defined by a displacement map

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

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

-**dissolve** <percent>  dissolve an image into another by the given percent

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

-**encoding** <type>  specify the font encoding

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

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

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

-**geometry** <width>x<height>{+}x{+}y{%}@{f}{<}{>}  preferred size and location of the Image window.

-**gravity** <type>  direction primitive gravitates to when annotating the image.
-help print usage instructions

-interlace <type> the type of interlacing scheme

-label <name> assign a label to an image

-matte store matte channel if the image has one

-monochrome transform the image to black and white

-negate replace every pixel with its complementary color

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

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

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

-resize <width>x<height>{%}{f}{<}{>} resize an image

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

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

-scene <value> set scene number

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

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

-stegano <offset> hide watermark within an image
-**stereo**  composite two images to create a stereo anaglyph

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

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

-**watermark** *<brightness>*<saturation>*  percent brightness and saturation of a watermark

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

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

### 3.6 Environment

**DISPLAY**  To get the default host, display number, and screen.

### 3.7 Authors

*John Cristy, ImageMagick Studio LLC,*  
*Glenn Randers-Pehrson, ImageMagick Studio LLC.*
4 Conjure

4.1 NAME

conjure - process a Magick Scripting Language (MSL) script

WARNING: Conjure and MSL are in very early development and the syn-
tax is subject to change!

4.2 Synopsis

conjure [ options ] script.msl [ [ options ] script.msl ]

4.3 Description

The Magick scripting language (MSL) will primarily benefit those that want to accomplish custom image processing tasks but do not wish to program, or those that do not have access to a Perl interpreter or a compiler. The interpreter is called conjure and here is an example script:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<image size="400x400">
  <read filename="image.gif" />
  <get width="base-width" height="base-height" />
  <resize geometry="%[dimensions]%" />
  <get width="width" height="height" />
  <print output="Image sized from %[base-width]x%[base-height] to %[width]x%[height].\n"/>
  <write filename="image.png"/>
</image>
```

56
invoked with

```
conjure -dimensions 400x400 incantation.msl
```

All operations will closely follow the key/value pairs defined in PerlMagick, unless otherwise noted.

**Conjure** is in the early stages of development as of April 2002.

This early announcement is to allow ImageMagick users to contribute ideas early in the process so when the scripting language is released it will be useful/stable from the get-go! If you want to contribute suggestions about the Magick scripting language (MSL), post them to magick-developers@imagemagick.org.

### 4.4 Options

Options are processed in command line order. Any option you specify on the command line remains in effect until it is explicitly changed by specifying the option again with a different effect, or if it is changed by a statement in the scripting language.

You can define your own keyword/value pairs on the command line. The script can then use this information when setting values by including %[keyword] in the string. For example, if you included “-dimensions 400x400” on the command line, as illustrated above, then any string containing “%[dimensions]” would have 400x400 substituted. The “%[string]” can be used either an entire string, such as geometry=“%[dimensions]” or as a part of a string such as filename=“%[basename].png”.

The keyword can be any string except for the following reserved strings (in any upper, lower, or mixed case variant): **debug**, **help**, and **verbose**, whose usage is described below.

The value can be any string. If either the keyword or the value contains white space or any symbols that have special meanings to your shell such as “#”, “[”, or “%”, enclose the string in quotation marks or use “´” to escape the white space and special symbols.

Keywords and values are case dependent. “Key”, “key”, and “KEY” would be three different keywords.

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

- **debug** enable debug printout

- **help** print usage instructions
-**verbose** print detailed information about the image

## 4.5 Magick Scripting Language

The Magick Scripting Language (MSL) presently defines the following elements and their attributes:

**<image>** Define a new image object. **</image>** destroys it. Because of this, if you wish to reference multiple “subimages” (aka pages or layers), you can embed one **image** element inside of another. For example:

```
<image>
<read filename="input.png" />
<get width="base-width" height="base-height" />
<image height="base-height" width="base-width">
<image />
<write filename="output.mng" />
</image>

<image size="400x400" />
```

**<group>** Define a new group of image objects. By default, images are only valid for the life of their **image** element.

```
<image> -- creates the image
..... -- do stuff with it
</image> -- dispose of the image
```

However, in a group, all images in that group will stay around for the life of the group:

```
<group> -- start a group
  <image> -- create an image
  ..... -- do stuff
  </image> -- NOOP
  <image> -- create another image
  ..... -- do more stuff
  </image>
  <write filename="image.mng" /> -- output
</group> -- dispose of both images
```

**<read>** Read a new image from a disk file.

```
<read filename="image.gif" />
```

To read two images use
<write> Write the image(s) to disk, either as a single multiple-image file or multiple ones if necessary.
<write filename=image.tiff />

<get> Get any attribute recognized by PerlMagick's GetAttribute() and stores it as an image attribute for later use. Currently only width and height are supported.
<get width="base-width" height="base-height" />
<print output="Image size is %[base-width]x%[base-height].\n" />

<set> background, bordercolor, clip-mask, colorspace, density, magick, mat-
tecolor, opacity. Set an attribute recognized by PerlMagick's GetAttribute().
<border> fill, geometry, height, width
<blur> radius, sigma
<charcoal> radius, sigma
<chop> geometry, height, width, x, y
<crop> geometry, height, width, x, y
<despeckle>
<emboss> radius, sigma
<enhance>
<equalize>
<flip>
<flop>
<frame> fill, geometry, height, width, x, y, inner, outer
<get> height, width
<image> background, color, id, size
<magnify>
<minify>
<normalize>
<print output
<read>
<resize> blur, filter, geometry, height, width
<roll> geometry, x, y
<rotate> degrees
<sample> geometry, height, width
<scale> geometry, height, width
<sharpen> radius, sigma
<shave> geometry, height, width
<shear> x, y
<solarize> threshold
<spread> radius
<stegano> image
<stereo> image
<swirl> degrees
<texture> image
<threshold> threshold
<transparent> color
<trim>

4.6 Authors

John Cristy, ImageMagick Studio LLC,
Glenn Randers-Pehrson, ImageMagick Studio LLC,
Leonard Rosenthal, ImageMagick Studio LLC.
5 Convert

5.1 NAME

convert - convert an image or sequence of images

5.2 Synopsis

convert [ options ... ] input_file output_file

5.3 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 ImageMagick(1).

5.4 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 downsampled to 120x120, allowing it to run faster by avoiding returning a full-resolution image. The ‘-resize 120x120’ specifies the desired dimensions of the output image. It will be scaled so its largest dimension...
is 120 pixels. The ‘+profile “*”’ removes any ICM, EXIF, IPTC, or other profiles that might be present in the input and aren’t needed in the thumbnail.

To convert a **MIFF** image of a cockatoo to a SUN raster image, use:

```
convert cockatoo.miff sun:cockatoo.ras
```

To convert a multi-page *PostScript* document to individual FAX pages, use:

```
convert -monochrome document.ps fax:page
```

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.tiff 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.tiff
convert img0009.pcd[4] image.tiff
```

To create a visual image directory of all your JPEG images, use:

```
convert 'vid:*.jpg' directory.miff
```

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 to use with Mosaic(1), use:

```
convert -mattecolor "#697B8F" -frame 6x6 bird.jpg icon.png
```

To create a MNG animation from a DNA molecule sequence, use:

```
convert -delay 20 dna.* dna.mng
```
5.5 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 \(-width\) \(x\)-\(height\) \(\{\pm\} \{\pm\} \{\pm\} \{\%\}\) 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 \(-width\) \(x\)-\(height\) \(\{\pm\} \{\pm\} \{\pm\} \{\%\}\) 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 font 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>{+-}<-y>{+-}{%}{f}{<}>{> } 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>{%}@{f}<{}>`  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

-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
-treedepth <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).

5.6 Segmentation

Use -segment to segment an image by analyzing the histograms of the color components and identifying units that are homogeneous with the fuzzy c-means technique. The scale-space filter analyzes the histograms of the three color components of the image and identifies a set of classes. The extents of each class is used to coarsely segment the image with thresholding. The color associated with each class is determined by the mean color of all pixels within the extents of a particular class. Finally, any unclassified pixels are assigned to the closest class with the fuzzy c-means technique.

The fuzzy c-Means algorithm can be summarized as follows:

Build a histogram, one for each color component of the image.
For each histogram, successively apply the scale-space filter and build an interval tree of zero crossings in the second derivative at each scale. Analyze this scale-space “fingerprint” to determine which peaks or valleys in the histogram are most predominant.

The fingerprint defines intervals on the axis of the histogram. Each interval contains either a minima or a maxima in the original signal. If each color component lies within the maxima interval, that pixel is considered “classified” and is assigned an unique class number.

Any pixel that fails to be classified in the above thresholding pass is classified using the fuzzy c-Means technique. It is assigned to one of the classes discovered in the histogram analysis phase.

The fuzzy c-Means technique attempts to cluster a pixel by finding the local minima of the generalized within group sum of squared error objective function. A pixel is assigned to the closest class of which the fuzzy membership has a maximum value.


5.7 Environment

DISPLAY To get the default host, display number, and screen.

5.8 Authors

John Cristy, ImageMagick Studio LLC,
Glenn Randers-Pehrson, ImageMagick Studio LLC.
6 Display

6.1 NAME

display - display an image on any workstation running X

6.2 Synopsis

display [ options ...] file [options...]file

6.3 Description

Display is a machine architecture independent image processing and display program. It can display an image on any workstation screen running an X server. **Display** can read and write many of the more popular image formats (e.g. JPEG, TIFF, PNM, Photo CD, etc.).

With **display**, you can perform these functions on an image:

- load an image from a file
- display the next image
- display the former image
- display a sequence of images as a slide show
- write the image to a file
- print the image to a *PostScript* printer
- delete the image file
- create a Visual Image Directory
- select the image to display by its thumbnail rather than name
- undo last image transformation
- copy a region of the image
paste a region to the image
restore the image to its original size
refresh the image
half the image size
double the image size
resize the image
crop the image
cut the image
flop image in the horizontal direction
flip image in the vertical direction
rotate the image 90 degrees clockwise
rotate the image 90 degrees counter-clockwise
rotate the image
shear the image
roll the image
trim the image edges
inverted the colors of the image
vary the color brightness
vary the color saturation
vary the image hue
gamma correct the image
sharpen the image contrast
dull the image contrast
perform histogram equalization on the image
perform histogram normalization on the image
negate the image colors
convert the image to grayscale
set the maximum number of unique colors in the image
reduce the speckles within an image
eliminate peak noise from an image
detect edges within the image
emboss an image
segment the image by color
simulate an oil painting
simulate a charcoal drawing
annotate the image with text
draw on the image
edit an image pixel color
edit the image matte information
composite an image with another
add a border to the image
surround image with an ornamental border
apply image processing techniques to a region of interest
display information about the image
zoom a portion of the image
show a histogram of the image
display image to background of a window
6.4 Examples

To scale an image of a cockatoo to exactly 640 pixels in width and 480 pixels in height and position the window at location (200,200), use:

\[
\text{display -geometry 640x480+200+200! cockatoo.miff}
\]

To display an image of a cockatoo without a border centered on a backdrop, use:

\[
\text{display +borderwidth -backdrop cockatoo.miff}
\]

To tile a slate texture onto the root window, use:

\[
\text{display -size 1280x1024 -window root slate.png}
\]

To display a visual image directory of all your JPEG images, use:

\[
\text{display ’vid:*.jpg’}
\]

To display a MAP image that is 640 pixels in width and 480 pixels in height with 256 colors, use:

\[
\text{display -size 640x480+256 cockatoo.map}
\]

To display an image of a cockatoo specified with a World Wide Web (WWW) uniform resource locator (URL), use:

\[
\text{display ftp://wizards.dupont.com/images/cockatoo.jpg}
\]

To display histogram of an image, use:

\[
\text{convert file.jpg HISTOGRAM:- | display -}
\]
6.5 Options

Options are processed in command line order. Any option you specify on the command line remains in effect until it is explicitly changed by specifying the option again with a different effect. For example to display three images, the first with 32 colors, the second with an unlimited number of colors, and the third with only 16 colors, use:

```
display -colors 32 cockatoo.miff -noop duck.miff
-colors 16 macaw.miff
```

Display options can appear on the command line or in your X resources file. See X(1). Options on the command line supersede values specified in your X resources file.

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

-**backdrop** <color> display the image centered on a backdrop.

-**background** <color> the background color

-**border** <width>x<height> surround the image with a border of color

-**bordercolor** <color> the border color

-**borderwidth** <geometry> the border width

-**cache** <threshold> megabytes of memory available to the pixel cache

-**colormap** <type> define the colormap type

-**colors** <value> preferred number of colors in the image

-**colorspace** <value> the type of colorspace

-**comment** <string> annotate an image with a comment
-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

-debug  enable debug printout

-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

-edge <radius>  detect edges within an image

-endian <type>  specify endianness (MSB or LSB) of output image

-enhance  apply a digital filter to enhance a noisy image

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

-flip  create a “mirror image”
-flopl create a “mirror image”

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

-foreground <color> define the foreground color

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

-gamma <value> level of gamma correction

-geometry <width>x<height> {+} <x> {+} <y> {=} <@} {=} <> preferred size and location of the Image window.

-help print usage instructions

-iconGeometry <geometry> specify the icon geometry

-iconic iconic animation

-immutable make image immutable

-interlace <type> the type of interlacing scheme

-label <name> assign a label to an image

-magnify <factor> magnify the image

-map <type> display image using this type.

-matte store matte channel if the image has one

-mattecolor <color> specify the color to be used with the -frame option
-`monochrome` transform the image to black and white

-`name` name an image

-`negate` replace every pixel with its complementary color

-`noop` NOOP (no option)

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

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

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

-`remote` perform a remote operation

-`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.

-`scenes` `<value-value>` range of image scene numbers to read

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

-`shared_memory` use shared memory

-`sharpen` `<radius>x<sigma>` sharpen the image
-size width \( \times \) height \{ +offset \} width and height of the image

-text_font name font for writing fixed-width text

-texture filename name of texture to tile onto the image background

-title string assign title to displayed image [animate, display, montage]

-treedepth value tree depth for the color reduction algorithm

-trim trim an image

-update seconds detect when image file is modified and redisplay.

-use pixmap use the pixmap

-verbose print detailed information about the image

-visual type animate images using this X visual type

-window id make image the background of a window

-window_group specify the window group

-write filename write the image to a file [display]

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

### 6.6 Mouse Buttons

The effects of each button press is described below. Three buttons are required. If you have a two button mouse, button 1 and 3 are returned. Press ALT and button 3 to simulate button 2.
1 Press this button to map or unmap the Command widget. See the next section for more information about the Command widget.

2 Press and drag to define a region of the image to magnify.

3 Press and drag to choose from a select set of `display(1)` commands. This button behaves differently if the image being displayed is a visual image directory.
   Choose a particular tile of the directory and press this button and drag to select a command from a pop-up menu. Choose from these menu items:

   Open
   Next
   Former
   Delete
   Update

   If you choose **Open**, the image represented by the tile is displayed. To return to the visual image directory, choose **Next** from the Command widget (refer to Command Widget). **Next** and **Former** moves to the next or former image respectively. Choose **Delete** to delete a particular image tile. Finally, choose **Update** to synchronize all the image tiles with their respective images. See montage and miff for more details.

6.7 Command Widget

The Command widget lists a number of sub-menus and commands. They are

**File**

Open...
Next
Former
Select...
Save...
Print...
Delete...
Canvas...
Visual Directory...
Quit

**Edit**

Undo
Redo
Cut
Copy
Paste

View
Half Size
Original Size
Double Size
Resize...
Apply
Refresh
Restore

Transform
Crop
Chop
Flop
Flip
Rotate Right
Rotate Left
Rotate...
Shear...
Roll...
Trim Edges

Enhance
Hue...
Saturation...
Brightness...
Gamma...
Spiff...
Dull
Equalize
Normalize
Negate
GRAYscale
Quantize...

Effects
Despeckle
Emboss
Reduce Noise
Add Noise
Sharpen...
Blur...
Threshold...
Edge Detect...
Menu items with a indented triangle have a sub-menu. They are represented above as the indented items. To access a sub-menu item, move the pointer to the appropriate menu and press button 1 and drag. When you find the desired sub-menu item, release the button and the command is executed. Move the pointer away from the sub-menu if you decide not to execute a particular command.
6.8 Keyboard Accelerators

Accelerators are one or two key presses that effect a particular command. The keyboard accelerators that display understands is:

- **Ctl+O**  Press to `<a href="#imlo">load</a>` an image from a file.
- **space**  Press to display the next image.

If the image is a multi-paged document such as a *PostScript* document, you can skip ahead several pages by preceding this command with a number. For example to display the fourth page beyond the current page, press `4space`.

- **backspace**  Press to display the former image.

If the image is a multi-paged document such as a *PostScript* document, you can skip behind several pages by preceding this command with a number. For example to display the fourth page preceding the current page, press `4n`.

- **Ctl-S**  Press to save the image to a file.
- **Ctl-P**  Press to print the image to a `<i>PostScript</i>` printer.
- **Ctl-D**  Press to delete an image file.
- **Ctl-N**  Press to create a blank canvas.
- **Ctl-Q**  Press to discard all images and exit program.
- **Ctl+Z**  Press to undo last image transformation.
- **Ctl+R**  Press to redo last image transformation.
- **Ctl-X**  Press to `<a href="#imcu">cut</a>` a region of the image.
- **Ctl-C**  Press to `<a href="#imco">copy</a>` a region of the image.
- **Ctl-V**  Press to `<a href="#impa">paste</a>` a region to the image.
- `<`  Press to halve the image size.
- `.`  Press to return to the original image size.
- `>`  Press to double the image size.
- `%`  Press to resize the image to a width and height you specify.
- **Cmd-A**  Press to make any image transformations permanent. By default, any image size transformations are applied to the original image to create the image displayed on the X server. However, the transformations are not permanent (i.e. the original image does not change size only the X image does). For example, if you press `>` the X image will appear to double in size, but the original image will in fact remain the same size. To force the
original image to double in size, press ">" followed by "Cmd-A".

@ Press to refresh the image window.
C Press to <a href="#imcr">crop</a> the image.
[ Press to <a href="#imch">chop</a> the image.
H Press to flop image in the horizontal direction.
V Press to flip image in the vertical direction.
/ Press to rotate the image 90 degrees clockwise.
\ Press to rotate the image 90 degrees counter-clockwise.
* Press to <a href="#imro">rotate</a> the image the number of degrees you specify.
S Press to shear the image the number of degrees you specify.
R Press to roll the image.
T Press to trim the image edges.
Shft-H Press to vary the color hue.
Shft-S Press to vary the color saturation.
Shft-L Press to vary the image brightness.
Shft-G Press to gamma correct the image.
Shft-C Press to spiff up the image contrast.
Shft-Z Press to dull the image contrast.  
= Press to perform histogram equalization on the image.
Shft-N Press to perform histogram normalization on the image.
Shft-~ Press to negate the colors of the image.
. Press to convert the image colors to gray.
Shft-# Press to set the maximum number of unique colors in the image.
F2 Press to reduce the speckles in an image.
F2 Press to emboss an image.
F4 Press to eliminate peak noise from an image.
F5 Press to add noise to an image.
F6 Press to sharpen an image.
F7 Press to blur image an image.
F8 Press to threshold the image.
F9 Press to detect edges within an image.
F10 Press to displace pixels by a random amount.
F11 Press to shade the image using a distant light source.
F12 Press to lighten or darken image edges to create a 3-D effect.
F13 Press to segment the image by color.
Meta-S Press to swirl image pixels about the center.
Meta-I Press to implode image pixels about the center.
Meta-W  Press to alter an image along a sine wave.
Meta-P  Press to simulate an oil painting.
Meta-C  Press to simulate a charcoal drawing.
Alt-X  Press to <a href="#imcomp">composite</a> the image with another.
Alt-A  Press to <a href="#iman">annotate</a> the image with text.
Alt-D  Press to <a href="#imdr">draw</a> a line on the image.
Alt-P  Press to <a href="#cole">edit</a> an image pixel color.
Alt-M  Press to <a href="#matt">edit</a> the image matte information.
Alt-X  Press to <a href="#imcomp">composite</a> the image with another.
Alt-A  Press to add a border to the image.
Alt-F  Press to add an ornamental frame to the image.
Alt-Shift! Press to add an image comment.
Ctl-A  Press to apply image processing techniques to a region of interest.
Shift-? Press to display information about the image.
Shift-+ Press to map the zoom image window.
Shift-P Press to preview an image enhancement, effect, or f/x.
F1    Press to display helpful information about the "display" utility.
Find  Press to browse documentation about ImageMagick.
1-9   Press to change the level of magnification.

Use the arrow keys to move the image one pixel up, down, left, or right within the magnify window. Be sure to first map the magnify window by pressing button 2.

Press ALT and one of the arrow keys to trim off one pixel from any side of the image.

6.9  X Resources

Display options can appear on the command line or in your X resource file. Options on the command line supersede values specified in your X resource file. See X(1) for more information on X resources.

Most display options have a corresponding X resource. In addition, display uses the following X resources:

**background (class Background)** Specifies the preferred color to use for the Image window background. The default is #ccc.

**borderColor (class BorderColor)** Specifies the preferred color to use for the Image window border. The default is #ccc.
**borderWidth (class BorderWidth)**  Specifies the width in pixels of the image window border. The default is 2.

**browseCommand (class browseCommand)**  Specifies the name of the preferred browser when displaying ImageMagick documentation. The default is `netscape %s`.

**confirmExit (class ConfirmExit)**  Display pops up a dialog box to confirm exiting the program when exiting the program. Set this resource to `False` to exit without a confirmation.

**displayGamma (class DisplayGamma)**  Specifies the gamma of the X server.

You can apply separate gamma values to the red, green, and blue channels of the image with a gamma value list delineated with slashes (i.e. `1.7/2.3/1.2`).

The default is 2.2.

**displayWarnings (class DisplayWarnings)**  Display pops up a dialog box whenever a warning message occurs. Set this resource to `False` to ignore warning messages.

**font (class FontList)**  Specifies the name of the preferred font to use in normal formatted text. The default is 14 point Helvetica.

**font[1-9] (class Font[1-9])**  Specifies the name of the preferred font to use when annotating the image window with text. The default fonts are fixed, variable, 5x8, 6x10, 7x13bold, 8x13bold, 9x15bold, 10x20, and 12x24.

**foreground (class Foreground)**  Specifies the preferred color to use for text within the image window. The default is black.

**gammaCorrect (class gammaCorrect)**  This resource, if true, will lighten or darken an image of known gamma to match the gamma of the display (see resource `displayGamma`). The default is True.

**geometry (class Geometry)**  Specifies the preferred size and position of the image window. It is not necessarily obeyed by all window managers.

Offsets, if present, are handled in `X(1)` style. A negative x offset is measured from the right edge of the screen to the right edge of the icon, and a negative y
offset is measured from the bottom edge of the screen to the bottom edge of the icon.

**iconGeometry (class IconGeometry)** Specifies the preferred size and position of the application when iconified. It is not necessarily obeyed by all window managers.

Offsets, if present, are handled in the same manner as in class Geometry.

**iconic (class Iconic)** This resource indicates that you would prefer that the application’s windows initially not be visible as if the windows had be immediately iconified by you. Window managers may choose not to honor the application’s request.

**magnify (class Magnify)** specifies an integral factor by which the image should be enlarged. The default is 3.

This value only affects the magnification window which is invoked with button number 3 after the image is displayed.

**matteColor (class MatteColor)** Specify the color of windows. It is used for the backgrounds of windows, menus, and notices. A 3D effect is achieved by using highlight and shadow colors derived from this color. Default value: #697B8F.

**name (class Name)** This resource specifies the name under which resources for the application should be found. This resource is useful in shell aliases to distinguish between invocations of an application, without resorting to creating links to alter the executable file name. The default is the application name.

**pen[1-9] (class Pen[1-9])** Specifies the color of the preferred font to use when annotating the image window with text. The default colors are black, blue, green, cyan, gray, red, magenta, yellow, and white.

**printCommand (class PrintCommand)** This command is executed whenever Print is issued. In general, it is the command to print PostScript to your printer. Default value: \texttt{lp -c -s \%i}.

**sharedMemory (class SharedMemory)** This resource specifies whether display should attempt use shared memory for pixmaps. ImageMagick must be compiled with shared memory support, and the display must support the MIT-SHM extension. Otherwise, this resource is ignored. The default is True.
**textFont (class textFont)** Specifies the name of the preferred font to use in fixed (typewriter style) formatted text. The default is 14 point Courier.

**title (class Title)** This resource specifies the title to be used for the image window. This information is sometimes used by a window manager to provide a header identifying the window. The default is the image file name.

**undoCache (class UndoCache)** Specifies, in mega-bytes, the amount of memory in the undo edit cache. Each time you modify the image it is saved in the undo edit cache as long as memory is available. You can subsequently undo one or more of these transformations. The default is 16 Megabytes.

**usePixmap (class UsePixmap)** Images are maintained as a XImage by default. Set this resource to True to utilize a server Pixmap instead. This option is useful if your image exceeds the dimensions of your server screen and you intend to pan the image. Panning is much faster with Pixmaps than with a XImage. Pixmaps are considered a precious resource, use them with discretion.

To set the geometry of the Magnify or Pan or window, use the geometry resource. For example, to set the Pan window geometry to 256x256, use:

\[
\text{display.pan.geometry: 256x256}
\]

### 6.10 Image Loading

To select an image to display, choose **Open** of the **File** sub-menu from the Command widget. A file browser is displayed. To choose a particular image file, move the pointer to the filename and press any button. The filename is copied to the text window. Next, press **Open** or press the **RETURN** key. Alternatively, you can type the image file name directly into the text window. To descend directories, choose a directory name and press the button twice quickly. A scrollbar allows a large list of filenames to be moved through the viewing area if it exceeds the size of the list area.

You can trim the list of file names by using shell globbing characters. For example, type `*.jpg` to list only files that end with `.jpg`. To select your image from the X server screen instead of from a file, Choose **Grab** of the **Open** widget.

### 6.11 Visual Image Directory

To create a Visual Image Directory, choose **Visual Directory** of the **File** sub-menu from the Command widget. A file browser is displayed. To create a Visual
Image Directory from all the images in the current directory, press Directory or press the RETURN key. Alternatively, you can select a set of image names by using shell globbing characters. For example, type *.jpg to include only files that end with .jpg. To descend directories, choose a directory name and press the button twice quickly. A scrollbar allows a large list of filenames to be moved through the viewing area if it exceeds the size of the list area.

After you select a set of files, they are turned into thumbnails and tiled onto a single image. Now move the pointer to a particular thumbnail and press button 3 and drag. Finally, select Open. The image represented by the thumbnail is displayed at its full size. Choose Next from the File sub-menu of the Command widget to return to the Visual Image Directory.

### 6.12 Image Cutting

Note that cut information for image window is not retained for colormapped X server visuals (e.g. StaticColor, StaticColor, GRAYScale, PseudoColor). Correct cutting behavior may require a TrueColor or DirectColor visual or a Standard Colormap.

To begin, press choose Cut of the Edit sub-menu from the Command widget. Alternatively, press F3 in the image window.

A small window appears showing the location of the cursor in the image window. You are now in cut mode. In cut mode, the Command widget has these options:

- Help
- Dismiss

To define a cut region, press button 1 and drag. The cut region is defined by a highlighted rectangle that expands or contracts as it follows the pointer. Once you are satisfied with the cut region, release the button. You are now in rectify mode. In rectify mode, the Command widget has these options:

- Cut
- Help
- Dismiss

You can make adjustments by moving the pointer to one of the cut rectangle corners, pressing a button, and dragging. Finally, press Cut to commit your copy region. To exit without cutting the image, press Dismiss.

### 6.13 Image Copying

To begin, press choose Copy of the Edit sub-menu from the Command widget. Alternatively, press F4 in the image window.
A small window appears showing the location of the cursor in the image window. You are now in copy mode. In copy mode, the Command widget has these options:

Help
Dismiss

To define a copy region, press button 1 and drag. The copy region is defined by a highlighted rectangle that expands or contracts as it follows the pointer. Once you are satisfied with the copy region, release the button. You are now in rectify mode. In rectify mode, the Command widget has these options:

Copy
Help
Dismiss

You can make adjustments by moving the pointer to one of the copy rectangle corners, pressing a button, and dragging. Finally, press Copy to commit your copy region. To exit without copying the image, press Dismiss.

6.14 Image Pasting

To begin, press choose Paste of the Edit sub-menu from the Command widget. Alternatively, press F5 in the image window.

A small window appears showing the location of the cursor in the image window. You are now in Paste mode. To exit immediately, press Dismiss. In Paste mode, the Command widget has these options:

Operators
  over
  in
  out
  atop
  xor
  plus
  minus
  add
  subtract
  difference
  multiply
  bumpmap
  replace
Help
Dismiss
Choose a composite operation from the **Operators** sub-menu of the Command widget. How each operator behaves is described below. *image window* is the image currently displayed on your X server and *image* is the image obtained with the File Browser widget.

**over** The result is the union of the two image shapes, with *image* obscuring *image window* in the region of overlap.

**in** The result is simply *image* cut by the shape of *image window*. None of the image data of *image window* is in the result.

**out** The resulting image is *image* with the shape of *image window* cut out.

**atop** The result is the same shape as *image window*, with *image* obscuring *image window* where the image shapes overlap. Note this differs from over because the portion of *image* outside *image window*’s shape does not appear in the result.

**xor** The result is the image data from both *image* and *image window* that is outside the overlap region. The overlap region is blank.

**plus** The result is just the sum of the image data. Output values are cropped to 255 (no overflow). This operation is independent of the matte channels.

**minus** The result of *image - image window*, with underflow cropped to zero. The matte channel is ignored (set to 255, full coverage).

**add** The result of *image + image window*, with overflow wrapping around (mod 256).

**subtract** The result of *image - image window*, with underflow wrapping around (mod 256). The add and subtract operators can be used to perform reversible transformations.

**difference** The result of abs(*image - image window*). This is useful for comparing two very similar images.

**multiply** The result of *image * image window*. This is useful for the creation of drop-shadows.
**bumpmap** The result of *image window* shaded by *window*.

**replace** The resulting image is *image window* replaced with *image*. Here the matte information is ignored.

The image compositor requires a matte, or alpha channel in the image for some operations. This extra channel usually defines a mask which represents a sort of a cookie-cutter for the image. This is the case when matte is 255 (full coverage) for pixels inside the shape, zero outside, and between zero and 255 on the boundary. If image does not have a matte channel, it is initialized with 0 for any pixel matching in color to pixel location (0,0), otherwise 255. See Matte Editing for a method of defining a matte channel.

Note that matte information for *image window* is not retained for colormapped X server visuals (e.g. StaticColor, StaticColor, GrayScale, PseudoColor). Correct compositing behavior may require a TrueColor or DirectColor visual or a Standard Colormap.

Choosing a composite operator is optional. The default operator is replace. However, you must choose a location to composite your image and press button 1. Press and hold the button before releasing and an outline of the image will appear to help you identify your location.

The actual colors of the pasted image is saved. However, the color that appears in *image window* may be different. For example, on a monochrome screen *image window* will appear black or white even though your pasted image may have many colors. If the image is saved to a file it is written with the correct colors. To assure the correct colors are saved in the final image, any PseudoClass image is promoted to DirectClass. To force a PseudoClass image to remain PseudoClass, use -colors.

### 6.15 Image Cropping

To begin, press choose **Crop** of the **Transform** submenu from the Command widget. Alternatively, press [ in the image window.

A small window appears showing the location of the cursor in the image window. You are now in crop mode. In crop mode, the Command widget has these options:

- **Help**
- **Dismiss**

To define a cropping region, press button 1 and drag. The cropping region is defined by a highlighted rectangle that expands or contracts as it follows the pointer. Once you are satisfied with the cropping region, release the button. You are now in rectify mode. In rectify mode, the Command widget has these options:
You can make adjustments by moving the pointer to one of the cropping rectangle corners, pressing a button, and dragging. Finally, press Crop to commit your cropping region. To exit without cropping the image, press Dismiss.

### 6.16 Image Chopping

An image is chopped interactively. There is no command line argument to chop an image. To begin, choose **Chop** of the **Transform** sub-menu from the Command widget. Alternatively, press ] in the Image window.

You are now in **Chop** mode. To exit immediately, press **Dismiss**. In Chop mode, the Command widget has these options:

**Direction**
- horizontal
- vertical

**Help**
**Dismiss**

If the you choose the horizontal direction (this is the default), the area of the image between the two horizontal endpoints of the chop line is removed. Otherwise, the area of the image between the two vertical endpoints of the chop line is removed.

Select a location within the image window to begin your chop, press and hold any button. Next, move the pointer to another location in the image. As you move a line will connect the initial location and the pointer. When you release the button, the area within the image to chop is determined by which direction you choose from the Command widget.

To cancel the image chopping, move the pointer back to the starting point of the line and release the button.

### 6.17 Image Rotation

Press the / key to rotate the image 90 degrees or to rotate -90 degrees. To interactively choose the degree of rotation, choose **Rotate...** of the **Transform** submenu from the Command Widget. Alternatively, press * in the image window.

A small horizontal line is drawn next to the pointer. You are now in rotate mode. To exit immediately, press **Dismiss**. In rotate mode, the Command widget has these options:
Choose a background color from the Pixel Color sub-menu. Additional background colors can be specified with the color browser. You can change the menu colors by setting the X resources pen1 through pen9.

If you choose the color browser and press Grab, you can select the background color by moving the pointer to the desired color on the screen and press any button.

Choose a point in the image window and press this button and hold. Next, move the pointer to another location in the image. As you move a line connects the initial location and the pointer. When you release the button, the degree of image rotation is determined by the slope of the line you just drew. The slope is relative to the direction you choose from the Direction sub-menu of the Command widget.

To cancel the image rotation, move the pointer back to the starting point of the line and release the button.

### 6.18 Image Segmentation

Choose Effects->Segment to segment an image by analyzing the histograms of the color components and identifying units that are homogeneous with the fuzzy c-means technique. The scale-space filter analyzes the histograms of the
three color components of the image and identifies a set of classes. The extents
of each class is used to coarsely segment the image with thresholding. The color
associated with each class is determined by the mean color of all pixels within
the extents of a particular class. Finally, any unclassified pixels are assigned to
the closest class with the fuzzy c-means technique. The fuzzy c-Means algorithm
can be summarized as follows:

Build a histogram, one for each color component of the image.
For each histogram, successively apply the scale-space filter and build an in-
terval tree of zero crossings in the second derivative at each scale. Analyze
this scale-space “fingerprint” to determine which peaks or valleys in the his-
togram are most predominant.
The fingerprint defines intervals on the axis of the histogram. Each interval
contains either a minima or a maxima in the original signal. If each color
component lies within the maxima interval, that pixel is considered “classi-
ified” and is assigned an unique class number.
Any pixel that fails to be classified in the above thresholding pass is classi-
ified using the fuzzy c-Means technique. It is assigned to one of the classes
discovered in the histogram analysis phase.

The fuzzy c-Means technique attempts to cluster a pixel by finding the local
minima of the generalized within group sum of squared error objective function.
A pixel is assigned to the closest class of which the fuzzy membership has a
maximum value.

For additional information see: Young Won Lim, Sang Uk Lee, “On The Color
Image Segmentation Algorithm Based on the Thresholding and the Fuzzy
c-Means Techniques”, Pattern Recognition, Volume 23, Number 9, pages 935-
952, 1990.

6.19 Image Annotation

An image is annotated interactively. There is no command line argument to an-
otate an image. To begin, choose Annotate of the Image Edit sub-menu from
the Command widget. Alternatively, press a in the image window.
A small window appears showing the location of the cursor in the image window.
You are now in annotate mode. To exit immediately, press Dismiss. In annotate
mode, the Command widget has these options:

Font Name
fixed
variable
5x8
6x10
7x13bold
Choose a font name from the **Font Name** sub-menu. Additional font names can be specified with the font browser. You can change the menu names by setting the X resources font1 through font9.
Choose a font color from the **Font Color** sub-menu. Additional font colors can be specified with the color browser. You can change the menu colors by setting the X resources pen1 through pen9.

If you select the color browser and press **Grab**, you can choose the font color by moving the pointer to the desired color on the screen and press any button.

If you choose to rotate the text, choose **Rotate Text** from the menu and select an angle. Typically you will only want to rotate one line of text at a time. Depending on the angle you choose, subsequent lines may end up overwriting each other.

Choosing a font and its color is optional. The default font is fixed and the default color is black. However, you must choose a location to begin entering text and press a button. An underscore character will appear at the location of the pointer. The cursor changes to a pencil to indicate you are in text mode. To exit immediately, press **Dismiss**.

In text mode, any key presses will display the character at the location of the underscore and advance the underscore cursor. Enter your text and once completed press **Apply** to finish your image annotation. To correct errors press **BACK SPACE**. To delete an entire line of text, press **DELETE**. Any text that exceeds the boundaries of the image window is automatically continued onto the next line.

The actual color you request for the font is saved in the image. However, the color that appears in your Image window may be different. For example, on a monochrome screen the text will appear black or white even if you choose the color red as the font color. However, the image saved to a file with **-write** is written with red lettering. To assure the correct color text in the final image, any *PseudoClass* image is promoted to *DirectClass* (see miff(5)). To force a *PseudoClass* image to remain *PseudoClass*, use **-colors**.

### 6.20 Image Compositing

An image composite is created interactively. **There is no command line argument to composite an image.** To begin, choose **Composite** of the **Image Edit** from the Command widget. Alternatively, press x in the Image window.

First a popup window is displayed requesting you to enter an image name. Press **Composite**, **Grab** or type a file name. Press **Cancel** if you choose not to create a composite image. When you choose **Grab**, move the pointer to the desired window and press any button.

If the **Composite** image does not have any matte information, you are informed and the file browser is displayed again. Enter the name of a mask image. The image is typically grayscale and the same size as the composite image. If the image is not grayscale, it is converted to grayscale and the resulting intensities are used as matte information.
A small window appears showing the location of the cursor in the image window. You are now in composite mode. To exit immediately, press Dismiss. In composite mode, the Command widget has these options:

**Operators**
- over
- in
- out
- atop
- xor
- plus
- minus
- add
- subtract
- difference
- bumpmap
- replace

**Blend**

**Displace**

**Help**

**Dismiss**

Choose a composite operation from the Operators sub-menu of the Command widget. How each operator behaves is described below. image window is the image currently displayed on your X server and image is the image obtained over

The result is the union of the two image shapes, with image obscuring image window in the region of overlap.

in

The result is simply image cut by the shape of image window. None of the image data of image window is in the result.

out

The resulting image is image with the shape of image window cut out.

atop

The result is the same shape as image window, with image obscuring image window where the image shapes overlap. Note this differs from over because the portion of image outside image window’s shape does not appear in the result.

xor

The result is the image data from both image and image window that is outside the overlap region. The overlap region is blank.
**plus** The result is just the sum of the image data. Output values are cropped to 255 (no overflow). This operation is independent of the matte channels.

**minus** The result of image - image window, with underflow cropped to zero. The matte channel is ignored (set to 255, full coverage).

**add** The result of image + image window, with overflow wrapping around (mod 256).

**subtract** The result of image - image window, with underflow wrapping around (mod 256). The add and subtract operators can be used to perform reversible transformations.

**difference** The result of abs(image - image window). This is useful for comparing two very similar images.

**bumpmap** The result of image window shaded by window.

**replace** The resulting image is image window replaced with image. Here the matte information is ignored.

The image compositor requires a matte, or alpha channel in the image for some operations. This extra channel usually defines a mask which represents a sort of a cookie-cutter for the image. This is the case when matte is 255 (full coverage) for pixels inside the shape, zero outside, and between zero and 255 on the boundary. If image does not have a matte channel, it is initialized with 0 for any pixel matching in color to pixel location (0,0), otherwise 255. See Matte Editing for a method of defining a matte channel.

If you choose **blend**, the composite operator becomes **over**. The image matte channel percent transparency is initialized to factor. The image window is initialized to $(100 \text{- factor})$. Where factor is the value you specify in the Dialog widget.

**Displace** shifts the image pixels as defined by a displacement map. With this option, image is used as a displacement map. Black, within the displacement map, is a maximum positive displacement. White is a maximum negative displacement and middle gray is neutral. The displacement is scaled to determine the pixel shift. By default, the displacement applies in both the horizontal and vertical directions. However, if you specify mask, image is the horizontal X displacement and mask the vertical Y displacement.

Note that matte information for image window is not retained for colormapped X server visuals (e.g. StaticColor, StaticColor, GrayScale, PseudoColor). Correct compositing behavior may require a TrueColor or DirectColor visual or a Standard Colormap.
Choosing a composite operator is optional. The default operator is replace. However, you must choose a location to composite your image and press button 1. Press and hold the button before releasing and an outline of the image will appear to help you identify your location.

The actual colors of the composite image is saved. However, the color that appears in image window may be different. For example, on a monochrome screen Image window will appear black or white even though your composited image may have many colors. If the image is saved to a file it is written with the correct colors. To assure the correct colors are saved in the final image, any PseudoClass image is promoted to DirectClass (see miff). To force a PseudoClass image to remain PseudoClass, use -colors.

6.21 Color Editing

Changing the the color of a set of pixels is performed interactively. There is no command line argument to edit a pixel. To begin, choose Color from the Image Edit submenu of the Command widget. Alternatively, press c in the image window.

A small window appears showing the location of the cursor in the image window. You are now in color edit mode. To exit immediately, press Dismiss. In color edit mode, the Command widget has these options:

Method
- point
- replace
- floodfill
- reset

Pixel Color
- black
- blue
- cyan
- green
- gray
- red
- magenta
- yellow
- white
- Browser...

Border Color
- black
- blue
- cyan
- green
- gray
Choose a color editing method from the Method sub-menu of the Command widget. The point method recolors any pixel selected with the pointer unless the button is released. The replace method recolors any pixel that matches the color of the pixel you select with a button press. Floodfill recolors any pixel that matches the color of the pixel you select with a button press and is a neighbor. Whereas filltoborder changes the matte value of any neighbor pixel that is not the border color. Finally reset changes the entire image to the designated color.

Next, choose a pixel color from the Pixel Color sub-menu. Additional pixel colors can be specified with the color browser. You can change the menu colors by setting the X resources pen1 through pen9.

Now press button 1 to select a pixel within the Image window to change its color. Additional pixels may be recolored as prescribed by the method you choose. additional pixels by increasing the Delta value.

If the Magnify widget is mapped, it can be helpful in positioning your pointer within the image (refer to button 2). Alternatively you can select a pixel to recolor from within the Magnify widget. Move the pointer to the Magnify widget and position the pixel with the cursor control keys. Finally, press a button to recolor the selected pixel (or pixels).

The actual color you request for the pixels is saved in the image. However, the color that appears in your Image window may be different. For example, on a monochrome screen the pixel will appear black or white even if you choose the color red as the pixel color. However, the image saved to a file with -write is written with red pixels. To assure the correct color text in the final image, any PseudoClass image is promoted to DirectClass To force a PseudoClass image to remain PseudoClass, use -colors.
6.22 Matte Editing

Matte information within an image is useful for some operations such as image compositing. This extra channel usually defines a mask which represents a sort of a cookie-cutter for the image. This is the case when matte is 255 (full coverage) for pixels inside the shape, zero outside, and between zero and 255 on the boundary.

Setting the matte information in an image is done interactively. There is no command line argument to edit a pixel. To begin, and choose Matte of the Image Edit sub-menu from the Command widget.

Alternatively, press m in the image window.

A small window appears showing the location of the cursor in the image window. You are now in matte edit mode. To exit immediately, press Dismiss. In matte edit mode, the Command widget has these options:

Method
point
replace
floodfill
reset

Border Color
black
blue
cyran
green
gray
red
magenta
yellow
white
Browser...

Fuzz
0
2
4
8
16
Dialog...

Matte
Undo
Help
Dismiss
Choose a matte editing method from the Method sub-menu of the Command widget. The **point method** changes the matte value of the any pixel selected with the pointer until the button is released. The **replace method** changes the matte value of any pixel that matches the color of the pixel you select with a button press. **Floodfill** changes the matte value of any pixel that matches the color of the pixel you select with a button press and is a neighbor. Whereas **filtoborder** recolors any neighbor pixel that is not the border color. Finally **reset** changes the entire image to the designated matte value. Choose Matte Value and a dialog appears requesting a matte value. Enter a value between 0 and 255. This value is assigned as the matte value of the selected pixel or pixels. Now, press any button to select a pixel within the Image window to change its matte value. You can change the matte value of additional pixels by increasing the Delta value. The Delta value is first added then subtracted from the red, green, and blue of the target color. Any pixels within the range also have their matte value updated. If the Magnify widget is mapped, it can be helpful in positioning your pointer within the image (refer to button 2). Alternatively you can select a pixel to change the matte value from within the Magnify widget. Move the pointer to the Magnify widget and position the pixel with the cursor control keys. Finally, press a button to change the matte value of the selected pixel (or pixels). Matte information is only valid in a DirectClass image. Therefore, any PseudoClass image is promoted to DirectClass. Note that matte information for PseudoClass is not retained for colormap X server visuals (e.g. StaticColor, StaticColor, GrayScale, PseudoColor) unless you immediately save your image to a file (refer to Write). Correct matte editing behavior may require a TrueColor or DirectColor visual or a Standard Colormap.

### 6.23 Image Drawing

An image is drawn upon interactively. **There is no command line argument to draw on an image.** To begin, choose Draw of the Image Edit sub-menu from the Command widget. Alternatively, press d in the image window.

The cursor changes to a crosshair to indicate you are in draw mode. To exit immediately, press Dismiss. In draw mode, the Command widget has these options:

**Primitive**
- point
- line
- rectangle
- fill rectangle
- circle
- fill circle
- ellipse
- fill ellipse
- polygon
Choose a drawing primitive from the **Primitive** sub-menu.

Next, choose a color from the **Color** sub-menu. Additional colors can be specified with the color browser. You can change the menu colors by setting the X resources pen1 through pen9. The transparent color updates the image matte channel and is useful for image compositing.

If you choose the color browser and press **Grab**, you can select the primitive color by moving the pointer to the desired color on the screen and press any button. The transparent color updates the image matte channel and is useful for image compositing.

Choose a stipple, if appropriate, from the **Stipple** sub-menu. Additional stipples can be specified with the file browser. Stipples obtained from the file browser must be on disk in the X11 bitmap format.
Choose a line width, if appropriate, from the **Width** sub-menu. To choose a specific width select the **Dialog** widget.

Choose a point in the image window and press button 1 and hold. Next, move the pointer to another location in the image. As you move, a line connects the initial location and the pointer. When you release the button, the image is updated with the primitive you just drew. For polygons, the image is updated when you press and release the button without moving the pointer.

To cancel image drawing, move the pointer back to the starting point of the line and release the button.

### 6.24 Region of Interest

To begin, press choose Region of Interest of the Pixel Transform sub-menu from the Command widget. Alternatively, press R in the image window.

A small window appears showing the location of the cursor in the image window. You are now in region of interest mode. In region of interest mode, the Command widget has these options:

- **Help**
- **Dismiss**

To define a region of interest, press button 1 and drag. The region of interest is defined by a highlighted rectangle that expands or contracts as it follows the pointer. Once you are satisfied with the region of interest, release the button. You are now in apply mode. In apply mode the Command widget has these options:

- **File**
  - Save...
  - Print...
- **Edit**
  - Undo
  - Redo
- **Transform**
  - Flip
  - Flop
  - Rotate Right
  - Rotate Left
- **Enhance**
  - Hue...
  - Saturation...
  - Brightness...
  - Gamma...
  - Spiff
Dull
Equalize
Normalize
Negate
GRAYscale
Quantize...

Effects
Despeckle
Emboss
Reduce Noise
Add Noise
Sharpen...
Blur...
Threshold...
Edge Detect...
Spread...
Shade...
Raise...
Segment...

F/X
Solarize...
Swirl...
Implode...
Wave...
Oil Paint
Charcoal Draw...

Miscellany
Image Info
Zoom Image
Show Preview...
Show Histogram
Show Matte

Help
Dismiss

You can make adjustments to the region of interest by moving the pointer to one of the rectangle corners, pressing a button, and dragging. Finally, choose an image processing technique from the Command widget. You can choose more than one image processing technique to apply to an area. Alternatively, you can move the region of interest before applying another image processing technique. To exit, press Dismiss.
6.25 Image Panning

When an image exceeds the width or height of the X server screen, display maps a small panning icon. The rectangle within the panning icon shows the area that is currently displayed in the image window. To pan about the image, press any button and drag the pointer within the panning icon. The pan rectangle moves with the pointer and the image window is updated to reflect the location of the rectangle within the panning icon. When you have selected the area of the image you wish to view, release the button.

Use the arrow keys to pan the image one pixel up, down, left, or right within the image window.

The panning icon is withdrawn if the image becomes smaller than the dimensions of the X server screen.

6.26 User Preferences

Preferences affect the default behavior of display(1). The preferences are either true or false and are stored in your home directory as .displayrc:

- **display image centered on a backdrop**
  This backdrop covers the entire workstation screen and is useful for hiding other X window activity while viewing the image. The color of the backdrop is specified as the background color. Refer to X Resources for details.

- **confirm on program exit**
  Ask for a confirmation before exiting the display(1) program.

- **correct image for display gamma**
  If the image has a known gamma, the gamma is corrected to match that of the X server (see the X Resource displayGamma).

- **apply Floyd/Steinberg error diffusion to image**
  The basic strategy of dithering is to trade intensity resolution for spatial resolution by averaging the intensities of several neighboring pixels. Images which suffer from severe contouring when reducing colors can be improved with this preference.

- **use a shared colormap for colormapped X visuals**
  This option only applies when the default X server visual is PseudoColor or GRAYScale. Refer to -visual for more details. By default, a shared colormap is allocated. The image shares colors with other X clients. Some image colors could be approximated, therefore your image may look very different than intended. Otherwise the image colors appear exactly as they are defined. However, other clients may go technicolor when the image colormap is installed.

- **display images as an X server pixmap**
Images are maintained as a XImage by default. Set this resource to True to utilize a server Pixmap instead. This option is useful if your image exceeds the dimensions of your server screen and you intend to pan the image. Panning is much faster with Pixmaps than with a XImage. Pixmaps are considered a precious resource, use them with discretion.

6.27 Environment

DISPLAY To get the default host, display number, and screen.

6.28 Acknowledgements

The MIT X Consortium for making network transparent graphics a reality.

Peder Langlo, Hewlett Packard, Norway, made hundreds of suggestions and bug reports. Without Peder, ImageMagick would not be nearly as useful as it is today.

Rod Bogart and John W. Peterson, University of Utah. Image compositing is loosely based on rlecomp of the Utah Raster Toolkit.

Michael Halle, Spatial Imaging Group at MIT, for the initial implementation of Alan Paeth’s image rotation algorithm.

David Pensak, ImageMagick Studio, for providing a computing environment that made this program possible.

Paul Raveling, USC Information Sciences Institute. The spatial subdivision color reduction algorithm is based on his Img software.

6.29 Authors

John Cristy, ImageMagick Studio LLC,
Glenn Randers-Pehrson, ImageMagick Studio LLC.
7 Identify

7.1 NAME

identify - describe an image or image sequence.

7.2 Synopsis

identify file [ file ... ]

7.3 Description

Identify describes the format and characteristics of one or more image files. It will also report if an image is incomplete or corrupt. The information displayed includes the scene number, the file name, the width and height of the image, whether the image is colormapped or not, the number of colors in the image, the number of bytes in the image, the format of the image (JPEG, PNM, etc.), and finally the number of seconds it took to read and process the image. An example line output from identify follows:

    images/aquarium.miff 640x480 PseudoClass 256c
                        308135b MIFF 1s

    Image: images/aquarium.miff
    class: PseudoClass
    colors: 256
    signature: eb5dca81dd93ae7e6fae99a527eb5dca8...

If -verbose is set, expect additional output including any image comment:
7.4 Options

Options are processed in command line order. Any option you specify on the command line remains in effect for the set of images immediately following, until the set is terminated by the appearance of any option or -noop.

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

-cache <threshold> megabytes of memory available to the pixel cache

d-debug enable debug printout

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

depth <value> depth of the image

-format <string> output formatted image characteristics

-help print usage instructions

-interlace <type> the type of interlacing scheme

-ping efficiently determine image characteristics

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

-size <width>x<height>{+offset} width and height of the image
-verbose  print detailed information about the image

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

7.5 Authors

John Cristy, ImageMagick Studio LLC,
Glenn Randers-Pehrson, ImageMagick Studio LLC.
8 Import

8.1 NAME

import - capture some or all of an X server screen and save the image to a file.

8.2 Synopsis

import [ options ... ] file

8.3 Description

Import reads an image from any visible window on an X server and outputs it as an image file. You can capture a single window, the entire screen, or any rectangular portion of the screen. Use display for redisplay, printing, editing, formatting, archiving, image processing, etc. of the captured image.

The target window can be specified by id, name, or may be selected by clicking the mouse in the desired window. If you press a button and then drag, a rectangle will form which expands and contracts as the mouse moves. To save the portion of the screen defined by the rectangle, just release the button. The keyboard bell is rung once at the beginning of the screen capture and twice when it completes.

8.4 Examples

To select an X window or an area of the screen with the mouse and save it in the MIFF image format to a file entitled window.miff, use:

import window.miff
To select an X window or an area of the screen with the mouse and save it in the Encapsulated PostScript format to include in another document, use:

```plaintext
import figure.eps
```

To capture the entire X server screen in the JPEG image format in a file entitled root.jpeg, without using the mouse, use:

```plaintext
import -window root root.jpeg
```

To capture the 512x256 area at the upper right corner of the X server screen in the PNG image format in a well-compressed file entitled corner.png, without using the mouse, use:

```plaintext
import -window root -crop 512x256-0+0 -quality 90 corner.png
```

### 8.5 Options

Options are processed in command line order. Any option you specify on the command line remains in effect until it is explicitly changed by specifying the option again with a different effect.

Import options can appear on the command line or in your X resources file. See X(1). Options on the command line supersede values specified in your X resources file.

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

- **-bordercolor** `<color>` the border color

- **-cache** `<threshold>` megabytes of memory available to the pixel cache

- **-colors** `<value>` preferred number of colors in the image

- **-colorspace** `<value>` the type of colorspace

- **-comment** `<string>` annotate an image with a comment

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

-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

-descend obtain image by descending window hierarchy

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

-dispose <method> GIF disposal method

-dither apply Floyd/Steinberg error diffusion to the image

-encoding <type> specify the font encoding

-endian <type> specify endianness (MSB or LSB) of output image

-frame include the X window frame in the imported image

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

-help print usage instructions

-interlace <type> the type of interlacing scheme

-label <name> assign a label to an image

-monochrome transform the image to black and white
-**negate** replace every pixel with its complementary color

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

-**pause** `<seconds>` pause between snapshots [import]

-**ping** efficiently determine image characteristics

-**pointsize** `<value>` pointsize of the PostScript, OPTION1, or TrueType font

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

-**resize** `<width>x<height>{%}{@}<{}>` resize an image

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

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

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

-**screen** specify the screen to capture

-**silent** operate silently

-**snaps** `<value>` number of screen snapshots

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

-**trim** trim an image

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

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

DISPLAY

8.7 Authors

John Cristy, ImageMagick Studio LLC,
Glenn Randers-Pehrson, ImageMagick Studio LLC.
9  Mogrify

9.1  NAME

mogrify - mogrify an image

9.2  Synopsis

mogrify [ options ... ] file ...

9.3  Description

Mogrify transforms an image or a sequence of images. These transforms include image scaling, image rotation, color reduction, and others. The transmogrified image overwrites the original image, unless an option such as -format causes the output filename to be different from the input filename.

The graphics formats supported by mogrify are listed in ImageMagick(1).

9.4  Examples

To convert all the TIFF files in a particular directory to JPEG, use:

    mogrify -format jpeg *.tiff

To convert a directory full of JPEG images to thumbnails, use:

    convert -size 120x120 *.jpg -resize 120x120 +profile "*"
In this example, ‘-i size 120x120’ gives a hint to the JPEG decoder that
the images are going to be downscaled to 120x120, allowing it to run faster by
avoiding returning full-resolution images. The ‘-resize 120x120’ specifies the desired dimensions of the output images. It will be scaled so its largest
dimension is 120 pixels. The ‘+profile "*"’ removes any ICM, EXIF,
IPTC, or other profiles that might be present in the input and aren’t needed in
the thumbnails.

To scale an image of a cockatoo to exactly 640 pixels in width and 480 pixels in
height, use:

    mogrify -resize 640x480! cockatoo.miff

### 9.5 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`.

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

- **-affine <matrix>** drawing transform matrix

- **-antialias** remove pixel aliasing

- **-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

- **-cache <threshold>** megabytes of memory available to the pixel cache

- **-channel <type>** the type of channel

- **-charcoal <factor>** simulate a charcoal drawing
-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
-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
-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 font 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

-**flip** create a “mirror image”

-**flop** create a “mirror image”

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

-**format** `<type>` the image format type

-**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

-**interlace** `<type>` the type of interlacing scheme

-**label** `<name>` assign a label to an image

-**level** `<value>` adjust the level of image contrast

-**linewidth** the line width for subsequent draw operations

-**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

-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>{%}{f}{l}{r}{g} size and location of an image canvas

-paint <radius> simulate an oil painting

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

-pointsize <value> pointsize of the PostScript, OPTION1, or TrueType font

-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>{%}{@}{f}{l}{r}{g} resize an image
-**roll**  \((\pm) \times x \pm y\) roll an image vertically or horizontally

-**rotate**  \(\langle degrees\rangle\{<\} \{>\} \) apply Paeth image rotation to the image

-**sample**  \(\langle geometry\rangle\) scale image with pixel sampling

-**sampling_factor**  \(\langle horizontal_factor\rangle \times \langle vertical_factor\rangle\) sampling factors used by JPEG or MPEG-2 encoder and YUV decoder/encoder.

-**scale**  \(\langle geometry\rangle\) scale the image.

-**scene**  \(\langle value\rangle\) set scene number

-**seed**  \(\langle value\rangle\) pseudo-random number generator seed value

-**segment**  \(\langle cluster_threshold\rangle \times \langle smoothing_threshold\rangle\) segment an image

-**shade**  \(\langle azimuth\rangle \times \langle elevation\rangle\) shade the image using a distant light source

-**sharpen**  \(\langle radius\rangle \times \langle sigma\rangle\) sharpen the image

-**shave**  \(\langle width\rangle \times \langle height\rangle\) shave pixels from the image edges

-**shear**  \(\langle x\, degrees\rangle \times \langle y\, degrees\rangle\) shear the image along the X or Y axis

-**size**  \(\langle width\rangle \times \langle height\rangle\{+\,offset\}\) width and height of the image

-**solarize**  \(\langle factor\rangle\) negate all pixels above the threshold level

-**spread**  \(\langle amount\rangle\) displace image pixels by a random amount

-**stroke**  \(\langle color\rangle\) color to use when stroking a graphic primitive
-**strokewidth** <value> set the stroke width

-**swirl** <degrees> swirl image pixels about the center

-**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

-**treedepth** <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

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

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

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

For a more detailed description of each option, see *ImageMagick(1)*.
9.6 Image Segmentation

Use `-segment` to segment an image by analyzing the histograms of the color components and identifying units that are homogeneous with the fuzzy c-means technique. The scale-space filter analyzes the histograms of the three color components of the image and identifies a set of classes. The extents of each class is used to coarsely segment the image with thresholding. The color associated with each class is determined by the mean color of all pixels within the extents of a particular class. Finally, any unclassified pixels are assigned to the closest class with the fuzzy c-means technique.

The fuzzy c-Means algorithm can be summarized as follows:

Build a histogram, one for each color component of the image.
For each histogram, successively apply the scale-space filter and build an interval tree of zero crossings in the second derivative at each scale. Analyze this scale-space “fingerprint” to determine which peaks or valleys in the histogram are most predominant.
The fingerprint defines intervals on the axis of the histogram. Each interval contains either a minima or a maxima in the original signal. If each color component lies within the maxima interval, that pixel is considered “classified” and is assigned an unique class number.
Any pixel that fails to be classified in the above thresholding pass is classified using the fuzzy c-Means technique. It is assigned to one of the classes discovered in the histogram analysis phase.

The fuzzy c-Means technique attempts to cluster a pixel by finding the local minima of the generalized within group sum of squared error objective function. A pixel is assigned to the closest class of which the fuzzy membership has a maximum value.


9.7 Environment

`DISPLAY` To get the default host, display number, and screen.

9.8 Authors

John Cristy, ImageMagick Studio LLC,
Glenn Randers-Pehrson, ImageMagick Studio LLC.
10 Montage

10.1 NAME

montage - create a composite image by combining several separate images

10.2 Synopsis

montage [ options ...] file [ [ options ...] file ...] output_file

10.3 Description

montage creates a composite image by combining several separate images. The images are tiled on the composite image with the name of the image optionally appearing just below the individual tile.

The composite image is constructed in the following manner. First, each image specified on the command line, except for the last, is scaled to fit the maximum tile size. The maximum tile size by default is 120x120. It can be modified with the -geometry command line argument or X resource. See Options for more information on command line arguments. See X(1) for more information on X resources. Note that the maximum tile size need not be a square.

Next the composite image is initialized with the color specified by the -background command line argument or X resource. The width and height of the composite image is determined by the title specified, the maximum tile size, the number of tiles per row, the tile border width and height, the image border width, and the label height. The number of tiles per row specifies how many images are to appear in each row of the composite image. The default is to have 5 tiles in each row and 4 tiles in each column of the composite. A specific value is specified with -tile. The tile border width and height, and the image border width

126
defaults to the value of the X resource `-borderwidth`. It can be changed with the `-borderwidth` or `-geometry` command line argument or X resource. The label height is determined by the font you specify with the `-font` command line argument or X resource. If you do not specify a font, a font is chosen that allows the name of the image to fit the maximum width of a tiled area. The label colors is determined by the `-background` and `-fill` command line argument or X resource. Note, that if the background and pen colors are the same, labels will not appear.

Initially, the composite image title is placed at the top if one is specified (refer to `-fill`). Next, each image is set onto the composite image, surrounded by its border color, with its name centered just below it. The individual images are left-justified within the width of the tiled area. The order of the images is the same as they appear on the command line unless the images have a scene keyword. If a scene number is specified in each image, then the images are tiled onto the composite in the order of their scene number. Finally, the last argument on the command line is the name assigned to the composite image. By default, the image is written in the MIFF format and can be viewed or printed with `display(1)`.

Note, that if the number of tiles exceeds the default number of 20 (5 per row, 4 per column), more than one composite image is created. To ensure a single image is produced, use `-tile` to increase the number of tiles to meet or exceed the number of input images.

Finally, to create one or more empty spaces in the sequence of tiles, use the “NULL:” image format.

Note, a composite MIFF image displayed to an X server with `display` behaves differently than other images. You can think of the composite as a visual image directory. Choose a particular tile of the composite and press a button to display it. See `display(1)` and `miff(5)`

### 10.4 Examples

To create a montage of a cockatoo, a parrot, and a hummingbird and write it to a file called birds, use:

```
montage cockatoo.miff parrot.miff hummingbird.miff birds.miff
```

To tile several bird images so that they are at most 256 pixels in width and 192 pixels in height, surrounded by a red border, and separated by 10 pixels of background color, use:

```
montage -geometry 256x192+10+10 -bordercolor red birds.* montage.miff
```
To create an unlabeled parrot image, 640 by 480 pixels, and surrounded by a border of black, use:

```
montage -geometry 640x480 -bordercolor black -label "" parrot.miff bird.miff
```

To create an image of an eagle with a textured background, use:

```
montage -texture bumps.jpg eagle.jpg eagle.png
```

To join several GIF images together without any extraneous graphics (e.g. no label, no shadowing, no surrounding tile frame), use:

```
montage +frame +shadow +label -tile 5x1 -geometry 50x50+0+0 *.png joined.png
```

### 10.5 Options

Any option you specify on the command line remains in effect for the group of images following it, until the group is terminated by the appearance of any option or `-noop`. For example, to make a montage of three images, the first with 32 colors, the second with an unlimited number of colors, and the third with only 16 colors, use:

```
montage -colors 32 cockatoo.1 -noop cockatoo.2 -colors 16 cockatoo.3 cockatoos.miff
```

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

- **-adjoin**: join images into a single multi-image file

- **-background <color>**: the background color

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

- **-bordercolor <color>**: the border color

- **-borderwidth <geometry>**: the border width

- **-cache <threshold>**: megabytes of memory available to the pixel cache
-chop \(< width \times height \{ +\} \times \{ +\} \times \{ \} \%\) remove pixels from the interior of an image

-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

-crop \(< width \times height \{ +\} \times \{ +\} \times \{ \} \%\) preferred size and location of the cropped image

-debug enable debug printout

-density \(< width \times height \) vertical and horizontal resolution in pixels of the image

-depth \(< value \) depth of the 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

-encoding \(< type \) specify the font encoding

-endian \(< type \) specify endianness (MSB or LSB) of output image
-fill <color>  color to use when filling a graphic primitive

-filter <type>  use this type of filter when resizing an 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

-gamma <value>  level of gamma correction

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

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

-help  print usage instructions

-interlace <type>  the type of interlacing scheme

-label <name>  assign a label to an image

-matte  store matte channel if the image has one

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

-mode <value>  mode of operation

-monochrome  transform the image to black and white

-noop  NOOP (no option)

-page <width>x<height>{+}{-}<x>{+}{-}<y>{%}{@}{l}{<}{>}  size and location of an image canvas
-pen <color> specify the pen color for drawing operations

-pointsize <value> pointsize of the PostScript, OPTION1, or TrueType font

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

-resize <width>x<height>{%}{@}{!}{<>} resize an image

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

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

-scenes <value-value> range of image scene numbers to read

-shadow <radius>x<sigma> shadow the montage

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

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

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

-strokewidth <value> set the stroke width

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

-tile <geometry> layout of images [montage]

-title <string> assign title to displayed image [animate, display, montage]

-transparent <color> make this color transparent within the image
**-treedepth <value>**  tree depth for the color reduction algorithm

**-trim** trim an image

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

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

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

## 10.6  X Resources

*Montage* options can appear on the command line or in your X resource file. Options on the command line supersede values specified in your X resource file. See *X(1)* for more information on X resources.

All *montage* options have a corresponding X resource. In addition, *montage* uses the following X resources:

**background (class Background)** background color

Specifies the preferred color to use for the composite image background. The default is #ccc.

**borderColor (class BorderColor)** border color

Specifies the preferred color to use for the composite image border. The default is #ccc.

**borderWidth (class BorderWidth)** border width

Specifies the width in pixels of the composite image border. The default is 2.

**font (class Font)** font to use

Specifies the name of the preferred font to use when displaying text within the composite image. The default is 9x15, fixed, or 5x8 determined by the composite image size.

**matteColor (class MatteColor)** color of the frame

Specify the color of an image frame. A 3D effect is achieved by using highlight and shadow colors derived from this color. The default value is #697B8F.
pen (class Pen) text color

Specifies the preferred color to use for text within the composite image. The default is black.

title (class Title) composite image title

This resource specifies the title to be placed at the top of the composite image. The default is not to place a title at the top of the composite image.

10.7 Environment

DISPLAY To get the default host, display number, and screen.

10.8 Acknowledgements

The MIT X Consortium for making network transparent graphics a reality.

Michael Halle, Spatial Imaging Group at MIT, for the initial implementation of Alan Paeth’s image rotation algorithm.

David Pensak, ImageMagick Studio, for providing a computing environment that made this program possible.

10.9 Authors

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