NAME

cjpeg − compress an image file to a JPEG file

SYNOPSIS

cjpeg [*options*] [*filename*]

DESCRIPTION

cjpeg compresses the named image file, or the standard input if no file is named, and produces a JPEG/JFIF file on the standard output. The currently supported input file formats are: PPM (PBMPLUS color format), PGM (PBMPLUS gray-scale format), BMP, Targa, and RLE (Utah Raster Toolkit format). (RLE is supported only if the URT library is available.)

OPTIONS

All switch names may be abbreviated; for example, **−grayscale** may be written **−gray** or **−gr**. Most of the "basic" switches can be abbreviated to as little as one letter. Upper and lower case are equivalent (thus **−BMP** is the same as **−bmp**). British spellings are also accepted (e.g., **−greyscale**), though for brevity these are not mentioned below.

The basic switches are:

−quality *N*
Scale quantization tables to adjust image quality. Quality is 0 (worst) to 100 (best); default is 75. (See below for more info.)

−grayscale
Create monochrome JPEG file from color input. Be sure to use this switch when compressing a grayscale BMP file, because **cjpeg** isn't bright enough to notice whether a BMP file uses only shades of gray. By saying **−grayscale**, you'll get a smaller JPEG file that takes less time to process.

−optimize
Perform optimization of entropy encoding parameters. Without this, default encoding parameters are used. **−optimize** usually makes the JPEG file a little smaller, but **cjpeg** runs somewhat slower and needs much more memory. Image quality and speed of decompression are unaffected by **−optimize**.

−progressive
Create progressive JPEG file (see below).

−targa Input file is Targa format. Targa files that contain an "identification" field will not be automatically recognized by **cjpeg**; for such files you must specify **−targa** to make **cjpeg** treat the input as Targa format. For most Targa files, you won't need this switch.

The **−quality** switch lets you trade off compressed file size against quality of the reconstructed image: the higher the quality setting, the larger the JPEG file, and the closer the output image will be to the original input. Normally you want to use the lowest quality setting (smallest file) that decompresses into something visually indistinguishable from the original image. For this purpose the quality setting should be between 50 and 95; the default of 75 is often about right. If you see defects at **−quality** 75, then go up 5 or 10 counts at a time until you are happy with the output image. (The optimal setting will vary from one image to another.)

−quality 100 will generate a quantization table of all 1's, minimizing loss in the quantization step (but there is still information loss in subsampling, as well as roundoff error). This setting is mainly of interest for experimental purposes. Quality values above about 95 are **not** recommended for normal use; the compressed file size goes up dramatically for hardly any gain in output image quality.

In the other direction, quality values below 50 will produce very small files of low image quality. Settings around 5 to 10 might be useful in preparing an index of a large image library, for example. Try **−quality** 2 (or so) for some amusing Cubist effects. (Note: quality values below about 25 generate 2-byte quantization tables, which are considered optional in the JPEG standard. **cjpeg** emits a warning message when you give such a quality value, because some other JPEG programs may be unable to decode the resulting file. Use **−baseline** if you need to ensure compatibility at low quality values.)

The **−progressive** switch creates a "progressive JPEG" file. In this type of JPEG file, the data is stored in multiple scans of increasing quality. If the file is being transmitted over a slow communications link, the decoder can use the first scan to display a low-quality image very quickly, and can then improve the display with each subsequent scan. The final image is exactly equivalent to a standard JPEG file of the same quality setting, and the total file size is about the same --- often a little smaller. **Caution:** progressive JPEG is not yet widely implemented, so many decoders will be unable to view a progressive JPEG file at all.

Switches for advanced users:

−dct int
Use integer DCT method (default).

−dct fast
Use fast integer DCT (less accurate).

−dct float
Use floating-point DCT method. The float method is very slightly more accurate than the int method, but is much slower unless your machine has very fast floating-point hardware. Also note that results of the floating-point method may vary slightly across machines, while the integer methods should give the same results everywhere. The fast integer method is much less accurate than the other two.

−restart *N*
Emit a JPEG restart marker every N MCU rows, or every N MCU blocks if "B" is attached to the number. **−restart 0** (the default) means no restart markers.

−smooth *N*
Smooth the input image to eliminate dithering noise. N, ranging from 1 to 100, indicates the strength of smoothing. 0 (the default) means no smoothing.

−maxmemory *N*
Set limit for amount of memory to use in processing large images. Value is in thousands of bytes, or millions of bytes if "M" is attached to the number. For example, **−max 4m** selects 4000000 bytes. If more space is needed, temporary files will be used.

−outfile *name*
Send output image to the named file, not to standard output.

−verbose
Enable debug printout. More **−v**'s give more output. Also, version information is printed at startup.

−debug
Same as **−verbose**.

The **−restart** option inserts extra markers that allow a JPEG decoder to resynchronize after a transmission error. Without restart markers, any damage to a compressed file will usually ruin the image from the point of the error to the end of the image; with restart markers, the damage is usually confined to the portion of the image up to the next restart marker. Of course, the restart markers occupy extra space. We recommend **−restart 1** for images that will be transmitted across unreliable networks such as Usenet.

The **−smooth** option filters the input to eliminate fine-scale noise. This is often useful when converting dithered images to JPEG: a moderate smoothing factor of 10 to 50 gets rid of dithering patterns in the input file, resulting in a smaller JPEG file and a better-looking image. Too large a smoothing factor will visibly blur the image, however.

Switches for wizards:

−baseline
Force baseline-compatible quantization tables to be generated. This clamps quantization values to 8 bits even at low quality settings. (This switch is poorly named, since it does not ensure that the output is actually baseline JPEG. For example, you can use **−baseline** and **−progressive** together.)

−qtables *file*
Use the quantization tables given in the specified text file.

−qslots *N[,...]*
: Select which quantization table to use for each color component.

−sample *HxV[,...]*
: Set JPEG sampling factors for each color component.

−scans *file*
: Use the scan script given in the specified text file.

The "wizard" switches are intended for experimentation with JPEG. If you don't know what you are doing, **don't use them**. These switches are documented further in the file wizard.doc.

EXAMPLES

This example compresses the PPM file foo.ppm with a quality factor of 60 and saves the output as foo.jpg:

cjpeg −quality *60 foo.ppm* **>** *foo.jpg*

HINTS

Color GIF files are not the ideal input for JPEG; JPEG is really intended for compressing full-color (24-bit) images. In particular, don't try to convert cartoons, line drawings, and other images that have only a few distinct colors. GIF works great on these, JPEG does not. If you want to convert a GIF to JPEG, you should experiment with **cjpeg**'s **−quality** and **−smooth** options to get a satisfactory conversion. **−smooth 10** or so is often helpful.

Avoid running an image through a series of JPEG compression/decompression cycles. Image quality loss will accumulate; after ten or so cycles the image may be noticeably worse than it was after one cycle. It's best to use a lossless format while manipulating an image, then convert to JPEG format when you are ready to file the image away.

The **−optimize** option to **cjpeg** is worth using when you are making a "final" version for posting or archiving. It's also a win when you are using low quality settings to make very small JPEG files; the percentage improvement is often a lot more than it is on larger files. (At present, **−optimize** mode is always selected when generating progressive JPEG files.)

ENVIRONMENT

JPEGMEM
: If this environment variable is set, its value is the default memory limit. The value is specified as described for the **−maxmemory** switch. **JPEGMEM** overrides the default value specified when the program was compiled, and itself is overridden by an explicit **−maxmemory**.

SEE ALSO

djpeg(1), **jpegtran**(1), **rdjpgcom**(1), **wrjpgcom**(1)
ppm(5), **pgm**(5)
Wallace, Gregory K. "The JPEG Still Picture Compression Standard", Communications of the ACM, April 1991 (vol. 34, no. 4), pp. 30-44.

AUTHOR

Independent JPEG Group

BUGS

Arithmetic coding is not supported for legal reasons.

GIF input files are no longer supported, to avoid the Unisys LZW patent. Use a Unisys-licensed program if you need to read a GIF file. (Conversion of GIF files to JPEG is usually a bad idea anyway.)

Not all variants of BMP and Targa file formats are supported.

The **−targa** switch is not a bug, it's a feature. (It would be a bug if the Targa format designers had not been clueless.)

Still not as fast as we'd like.

NAME

djpeg − decompress a JPEG file to an image file

SYNOPSIS

djpeg [*options*] [*filename*]

DESCRIPTION

djpeg decompresses the named JPEG file, or the standard input if no file is named, and produces an image file on the standard output. PBMPLUS (PPM/PGM), BMP, GIF, Targa, or RLE (Utah Raster Toolkit) output format can be selected. (RLE is supported only if the URT library is available.)

OPTIONS

All switch names may be abbreviated; for example, **−grayscale** may be written **−gray** or **−gr**. Most of the "basic" switches can be abbreviated to as little as one letter. Upper and lower case are equivalent (thus **−BMP** is the same as **−bmp**). British spellings are also accepted (e.g., **−greyscale**), though for brevity these are not mentioned below.

The basic switches are:

−colors *N*
: Reduce image to at most N colors. This reduces the number of colors used in the output image, so that it can be displayed on a colormapped display or stored in a colormapped file format. For example, if you have an 8-bit display, you'd need to reduce to 256 or fewer colors.

−quantize *N*
: Same as **−colors**. **−colors** is the recommended name, **−quantize** is provided only for backwards compatibility.

−fast
: Select recommended processing options for fast, low quality output. (The default options are chosen for highest quality output.) Currently, this is equivalent to **−dct fast −nosmooth −onepass −dither ordered**.

−grayscale
: Force gray-scale output even if JPEG file is color. Useful for viewing on monochrome displays; also, **djpeg** runs noticeably faster in this mode.

−scale *M/N*
: Scale the output image by a factor M/N. Currently the scale factor must be 1/1, 1/2, 1/4, or 1/8. Scaling is handy if the image is larger than your screen; also, **djpeg** runs much faster when scaling down the output.

−bmp
: Select BMP output format (Windows flavor). 8-bit colormapped format is emitted if **−colors** or **−grayscale** is specified, or if the JPEG file is gray-scale; otherwise, 24-bit full-color format is emitted.

−gif
: Select GIF output format. Since GIF does not support more than 256 colors, **−colors 256** is assumed (unless you specify a smaller number of colors).

−os2
: Select BMP output format (OS/2 1.x flavor). 8-bit colormapped format is emitted if **−colors** or **−grayscale** is specified, or if the JPEG file is gray-scale; otherwise, 24-bit full-color format is emitted.

−pnm
: Select PBMPLUS (PPM/PGM) output format (this is the default format). PGM is emitted if the JPEG file is gray-scale or if **−grayscale** is specified; otherwise PPM is emitted.

−rle
: Select RLE output format. (Requires URT library.)

−targa
: Select Targa output format. Gray-scale format is emitted if the JPEG file is gray-scale or if **−grayscale** is specified; otherwise, colormapped format is emitted if **−colors** is specified; otherwise, 24-bit full-color format is emitted.

Switches for advanced users:

−dct int
: Use integer DCT method (default).

−dct fast
Use fast integer DCT (less accurate).

−dct float
Use floating-point DCT method. The float method is very slightly more accurate than the int method, but is much slower unless your machine has very fast floating-point hardware. Also note that results of the floating-point method may vary slightly across machines, while the integer methods should give the same results everywhere. The fast integer method is much less accurate than the other two.

−dither fs
Use Floyd-Steinberg dithering in color quantization.

−dither ordered
Use ordered dithering in color quantization.

−dither none
Do not use dithering in color quantization. By default, Floyd-Steinberg dithering is applied when quantizing colors; this is slow but usually produces the best results. Ordered dither is a compromise between speed and quality; no dithering is fast but usually looks awful. Note that these switches have no effect unless color quantization is being done. Ordered dither is only available in **−onepass** mode.

−map *file*
Quantize to the colors used in the specified image file. This is useful for producing multiple files with identical color maps, or for forcing a predefined set of colors to be used. The *file* must be a GIF or PPM file. This option overrides **−colors** and **−onepass**.

−nosmooth
Use a faster, lower-quality upsampling routine.

−onepass
Use one-pass instead of two-pass color quantization. The one-pass method is faster and needs less memory, but it produces a lower-quality image. **−onepass** is ignored unless you also say **−colors** *N*. Also, the one-pass method is always used for gray-scale output (the two-pass method is no improvement then).

−maxmemory *N*
Set limit for amount of memory to use in processing large images. Value is in thousands of bytes, or millions of bytes if "M" is attached to the number. For example, **−max 4m** selects 4000000 bytes. If more space is needed, temporary files will be used.

−outfile *name*
Send output image to the named file, not to standard output.

−verbose
Enable debug printout. More **−v**'s give more output. Also, version information is printed at startup.

−debug
Same as **−verbose**.

EXAMPLES

This example decompresses the JPEG file foo.jpg, quantizes it to 256 colors, and saves the output in 8-bit BMP format in foo.bmp:

djpeg −colors 256 −bmp *foo.jpg* > *foo.bmp*

HINTS

To get a quick preview of an image, use the **−grayscale** and/or **−scale** switches. **−grayscale −scale 1/8** is the fastest case.

Several options are available that trade off image quality to gain speed. **−fast** turns on the recommended settings.

−dct fast and/or **−nosmooth** gain speed at a small sacrifice in quality. When producing a color-quantized image, **−onepass −dither ordered** is fast but much lower quality than the default behavior. **−dither none** may give acceptable results in two-pass mode, but is seldom tolerable in one-pass mode.

If you are fortunate enough to have very fast floating point hardware, **−dct float** may be even faster than **−dct fast**. But on most machines **−dct float** is slower than **−dct int**; in this case it is not worth using, because its theoretical accuracy advantage is too small to be significant in practice.

ENVIRONMENT

JPEGMEM

If this environment variable is set, its value is the default memory limit. The value is specified as described for the **−maxmemory** switch. **JPEGMEM** overrides the default value specified when the program was compiled, and itself is overridden by an explicit **−maxmemory**.

SEE ALSO

cjpeg(1), **jpegtran**(1), **rdjpgcom**(1), **wrjpgcom**(1)
ppm(5), **pgm**(5)
Wallace, Gregory K. "The JPEG Still Picture Compression Standard", Communications of the ACM, April 1991 (vol. 34, no. 4), pp. 30-44.

AUTHOR

Independent JPEG Group

BUGS

Arithmetic coding is not supported for legal reasons.

To avoid the Unisys LZW patent, **djpeg** produces uncompressed GIF files. These are larger than they should be, but are readable by standard GIF decoders.

Still not as fast as we’d like.

NAME

exifautotran − Transforms Exif files so that Orientation becomes 1

DESCRIPTION

exifautotran [list of files]

Take a list of files as input and transform then in place so that the Orientation becomes 1.

AUTHOR

Guido Vollbeding <guido@jpegclub.org>

SEE ALSO

jpegtran(1) jpegexiforient(1)

NAME

jpegexiforient − reads or writes the Exif Orientation Tag

SYNOPSIS

jpegexiforient [*switches*] *jpegfile*

DESCRIPTION

```
This is a utility program to get and set the Exif Orientation Tag.
It can be used together with jpegtran in scripts for automatic
orientation correction of digital camera pictures.

The Exif orientation value gives the orientation of the camera
relative to the scene when the image was captured.  The relation
of the '0th row' and '0th column' to visual position is shown as
below.

Value | 0th Row     | 0th Column
------+-------------+-----------
  1   | top         | left side
  2   | top         | rigth side
  3   | bottom      | rigth side
  4   | bottom      | left side
  5   | left side   | top
  6   | right side  | top
  7   | right side  | bottom
  8   | left side   | bottom

For convenience, here is what the letter F would look like if it were
tagged correctly and displayed by a program that ignores the orientation
tag:

  1       2       3      4

888888  888888      88  88
88          88      88  88
8888      8888    8888  8888
88          88      88  88
88          88  888888  888888

    5            6           7          8

8888888888  88                  88  8888888888
88  88      88  88          88  88      88  88
88          8888888888  8888888888          88
```

jpegexiforient output the Exif Orientation Tag in a JPEG Exif file. With the options -1 .. -8, it can also be used to set the tag.

OPTIONS

−−help display this help and exit

−−version
output version information and exit

−n Do not output the trailing newline

−1 .. **−8**
Set orientation value 1 .. 8

AUTHOR

Guido Vollbeding <guido@jpegclub.org>

SEE ALSO

jpegtran(1) exifautotran(1)

NAME

jpegtran − lossless transformation of JPEG files

SYNOPSIS

jpegtran [*options*] [*filename*]

DESCRIPTION

jpegtran performs various useful transformations of JPEG files. It can translate the coded representation from one variant of JPEG to another, for example from baseline JPEG to progressive JPEG or vice versa. It can also perform some rearrangements of the image data, for example turning an image from landscape to portrait format by rotation.

jpegtran works by rearranging the compressed data (DCT coefficients), without ever fully decoding the image. Therefore, its transformations are lossless: there is no image degradation at all, which would not be true if you used **djpeg** followed by **cjpeg** to accomplish the same conversion. But by the same token, **jpegtran** cannot perform lossy operations such as changing the image quality.

jpegtran reads the named JPEG/JFIF file, or the standard input if no file is named, and produces a JPEG/JFIF file on the standard output.

OPTIONS

All switch names may be abbreviated; for example, **−optimize** may be written **−opt** or **−o**. Upper and lower case are equivalent. British spellings are also accepted (e.g., **−optimise**), though for brevity these are not mentioned below.

To specify the coded JPEG representation used in the output file, **jpegtran** accepts a subset of the switches recognized by **cjpeg**:

−optimize
Perform optimization of entropy encoding parameters.

−progressive
Create progressive JPEG file.

−restart *N*
Emit a JPEG restart marker every N MCU rows, or every N MCU blocks if "B" is attached to the number.

−scans *file*
Use the scan script given in the specified text file.

See **cjpeg**(1) for more details about these switches. If you specify none of these switches, you get a plain baseline-JPEG output file. The quality setting and so forth are determined by the input file.

The image can be losslessly transformed by giving one of these switches:

−crop WxH+X+Y
Crop to a rectangular subarea of width W, height H starting at point X,Y. Allows the Width and or Height of the image to exceed the image.

−drop +X+Y
Drop another image. Overlay part of the source image at point width W + height H.

−flip horizontal
Mirror image horizontally (left-right).

−flip vertical
Mirror image vertically (top-bottom).

−perfect
Fails with an error if there is any loss durring the transformation.

−rotate 90
Rotate image 90 degrees clockwise.

−rotate 180
Rotate image 180 degrees.

−rotate 270
Rotate image 270 degrees clockwise (or 90 ccw).

−transpose
Transpose image (across UL-to-LR axis).

−transverse
Transverse transpose (across UR-to-LL axis).

The transpose transformation has no restrictions regarding image dimensions. The other transformations operate rather oddly if the image dimensions are not a multiple of the iMCU size (usually 8 or 16 pixels), because they can only transform complete blocks of DCT coefficient data in the desired way.

jpegtran's default behavior when transforming an odd-size image is designed to preserve exact reversibility and mathematical consistency of the transformation set. As stated, transpose is able to flip the entire image area. Horizontal mirroring leaves any partial iMCU column at the right edge untouched, but is able to flip all rows of the image. Similarly, vertical mirroring leaves any partial iMCU row at the bottom edge untouched, but is able to flip all columns. The other transforms can be built up as sequences of transpose and flip operations; for consistency, their actions on edge pixels are defined to be the same as the end result of the corresponding transpose-and-flip sequence.

For practical use, you may prefer to discard any untransformable edge pixels rather than having a strange-looking strip along the right and/or bottom edges of a transformed image. To do this, add the **−trim** switch:

−trim Drop non-transformable edge blocks.

Obviously, a transformation with **−trim** is not reversible, so strictly speaking **jpegtran** with this switch is not lossless. Also, the expected mathematical equivalences between the transformations no longer hold. For example, **−rot 270 -trim** trims only the bottom edge, but **−rot 90 -trim** followed by **−rot 180 -trim** trims both edges.

If you are only interested by perfect transformation, add the **−perfect** switch:

−perfect
Fails with an error if the transformation is not perfect. For example you may want to do

(jpegtran −rot 90 -perfect foo.jpg || djpeg foo.jpg| pnmflip −r90 | cjpeg)
to do a perfect rotation if available or an approximated one if not.

We also offer a lossless-crop option, which discards data outside a given image region but losslessly preserves what is inside. Like the rotate and flip transforms, lossless crop is restricted by the JPEG format: the upper left corner of the selected region must fall on an iMCU boundary. If this does not hold for the given crop parameters, we silently move the upper left corner up and/or left to make it so, simultaneously increasing the region dimensions to keep the lower right crop corner unchanged. (Thus, the output image covers at least the requested region, but may cover more.)

Note: **−perfect** and **lossless-crop** are enhancements from http://sylvana.net/jpegcrop/ that may not be available on non-Debian systems.

The image can be losslessly cropped by giving the switch:

−crop WxH+X+Y
Crop to a rectangular subarea of width W, height H starting at point X,Y.

Another not-strictly-lossless transformation switch is:

−grayscale
Force grayscale output.

This option discards the chrominance channels if the input image is YCbCr (ie, a standard color JPEG), resulting in a grayscale JPEG file. The luminance channel is preserved exactly, so this is a better method of reducing to grayscale than decompression, conversion, and recompression. This switch is particularly handy for fixing a monochrome picture that was mistakenly encoded as a color JPEG. (In such a case, the space savings from getting rid of the near-empty chroma channels won't be large; but the decoding time for a grayscale JPEG is substantially less than that for a color JPEG.)

jpegtran also recognizes these switches that control what to do with "extra" markers, such as comment blocks:

−copy none
Copy no extra markers from source file. This setting suppresses all comments and other excess baggage present in the source file.

−copy comments
Copy only comment markers. This setting copies comments from the source file, but discards any other inessential data.

−copy all
Copy all extra markers. This setting preserves miscellaneous markers found in the source file, such as JFIF thumbnails and Photoshop settings. In some files these extra markers can be sizable. See ’EXIF FILES’ for special tratement of EXIF markers.

−copy exif
This setting preserves the EXIF marker, commonly found in JPEG files produced by digital cameras, in addition to any comment markers. If there is an EXIF marker it is copied and the JFIF marker (incompatible with EXIF) is omitted. If there is no EXIF marker a JFIF one is emitted as usual. See ’EXIF FILES’ for special tratement of EXIF markers.

The default behavior is **−copy comments**. (Note: in IJG releases v6 and v6a, **jpegtran** always did the equivalent of **−copy none**.)

Additional switches recognized by jpegtran are:

−maxmemory *N*
Set limit for amount of memory to use in processing large images. Value is in thousands of bytes, or millions of bytes if "M" is attached to the number. For example, **−max 4m** selects 4000000 bytes. If more space is needed, temporary files will be used.

−outfile *name*
Send output image to the named file, not to standard output.

−verbose
Enable debug printout. More **−v**’s give more output. Also, version information is printed at startup.

−debug
Same as **−verbose**.

EXIF FILES

The EXIF variety of JPEG files, which are often produced by digital cameras, are recognized by jpegtran as EXIF files (i.e. not as JFIF, the usual variety of JPEG files). If the input file is recognized as EXIF (i.e., there is an EXIF marker and no JFIF marker) the ’-copy exif’ option is automatically turned on if ’-copy comments’, or no ’-copy’ option, was specified. Thus, unless ’-copy none’ is specified an EXIF file is kept as EXIF and not converted to JFIF.

If a geometrical transformation is applied (e.g., rotate, transpose) the EXIF width and height fields are set to the width and height of the output image. Furthermore, the orientation field is reset to one, meaning tha the orientation of the output image is upright (i.e. normal).

Note that an explicitely given ’-copy exif’ option will output an EXIF file if the input is an EXIF file that was saved as JFIF, and that the EXIF marker is still present. This option is useful for recovering EXIF files that where converted to JFIF by a non EXIF-aware software. Note however, that the data in the EXIF marker is not validated, unless a geometrical transformation is applied.

EXAMPLES

This example converts a baseline JPEG file to progressive form:

jpegtran −progressive *foo.jpg* > *fooprog.jpg*

This example rotates an image 90 degrees clockwise, discarding any unrotatable edge pixels:

jpegtran −rot 90 -trim *foo.jpg* > *foo90.jpg*

ENVIRONMENT

JPEGMEM

If this environment variable is set, its value is the default memory limit. The value is specified as described for the **−maxmemory** switch. **JPEGMEM** overrides the default value specified when the program was compiled, and itself is overridden by an explicit **−maxmemory**.

SEE ALSO

cjpeg(1), **djpeg**(1), **rdjpgcom**(1), **wrjpgcom**(1)

Wallace, Gregory K. "The JPEG Still Picture Compression Standard", Communications of the ACM, April 1991 (vol. 34, no. 4), pp. 30-44.

AUTHOR

Independent JPEG Group

BUGS

Arithmetic coding is not supported for legal reasons.

The transform options can't transform odd-size images perfectly. Use **−trim** or **−perfect** if you don't like the results.

The entire image is read into memory and then written out again, even in cases where this isn't really necessary. Expect swapping on large images, especially when using the more complex transform options.

NAME

rdjpgcom − display text comments from a JPEG file

SYNOPSIS

rdjpgcom [**−verbose**] [*filename*]

DESCRIPTION

rdjpgcom reads the named JPEG/JFIF file, or the standard input if no file is named, and prints any text comments found in the file on the standard output.

The JPEG standard allows "comment" (COM) blocks to occur within a JPEG file. Although the standard doesn't actually define what COM blocks are for, they are widely used to hold user-supplied text strings. This lets you add annotations, titles, index terms, etc to your JPEG files, and later retrieve them as text. COM blocks do not interfere with the image stored in the JPEG file. The maximum size of a COM block is 64K, but you can have as many of them as you like in one JPEG file.

OPTIONS

−verbose

Causes **rdjpgcom** to also display the JPEG image dimensions.

Switch names may be abbreviated, and are not case sensitive.

HINTS

rdjpgcom does not depend on the IJG JPEG library. Its source code is intended as an illustration of the minimum amount of code required to parse a JPEG file header correctly.

In **−verbose** mode, **rdjpgcom** will also attempt to print the contents of any "APP12" markers as text. Some digital cameras produce APP12 markers containing useful textual information. If you like, you can modify the source code to print other APPn marker types as well.

SEE ALSO

cjpeg(1), **djpeg**(1), **jpegtran**(1), **wrjpgcom**(1)

AUTHOR

Independent JPEG Group

NAME

wrjpgcom − insert text comments into a JPEG file

SYNOPSIS

wrjpgcom [**−replace**] [**−comment** *text*] [**−cfile** *name*] [*filename*]

DESCRIPTION

wrjpgcom reads the named JPEG/JFIF file, or the standard input if no file is named, and generates a new JPEG/JFIF file on standard output. A comment block is added to the file.

The JPEG standard allows "comment" (COM) blocks to occur within a JPEG file. Although the standard doesn't actually define what COM blocks are for, they are widely used to hold user-supplied text strings. This lets you add annotations, titles, index terms, etc to your JPEG files, and later retrieve them as text. COM blocks do not interfere with the image stored in the JPEG file. The maximum size of a COM block is 64K, but you can have as many of them as you like in one JPEG file.

wrjpgcom adds a COM block, containing text you provide, to a JPEG file. Ordinarily, the COM block is added after any existing COM blocks; but you can delete the old COM blocks if you wish.

OPTIONS

Switch names may be abbreviated, and are not case sensitive.

−replace
Delete any existing COM blocks from the file.

−comment *text*
Supply text for new COM block on command line.

−cfile *name*
Read text for new COM block from named file.

If you have only one line of comment text to add, you can provide it on the command line with **−comment**. The comment text must be surrounded with quotes so that it is treated as a single argument. Longer comments can be read from a text file.

If you give neither **−comment** nor **−cfile**, then **wrjpgcom** will read the comment text from standard input. (In this case an input image file name MUST be supplied, so that the source JPEG file comes from somewhere else.) You can enter multiple lines, up to 64KB worth. Type an end-of-file indicator (usually control-D) to terminate the comment text entry.

wrjpgcom will not add a COM block if the provided comment string is empty. Therefore **−replace −comment ""** can be used to delete all COM blocks from a file.

EXAMPLES

Add a short comment to in.jpg, producing out.jpg:

wrjpgcom −c *"View of my back yard" in.jpg* **>** *out.jpg*

Attach a long comment previously stored in comment.txt:

wrjpgcom *in.jpg* **<** *comment.txt* **>** *out.jpg*

or equivalently

wrjpgcom -cfile *comment.txt* **<** *in.jpg* **>** *out.jpg*

SEE ALSO

cjpeg(1), **djpeg**(1), **jpegtran**(1), **rdjpgcom**(1)

AUTHOR

Independent JPEG Group