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, GIF, 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 −**GIF** is the same as −**gif**). 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 GIF file, because **cjpeg** isn't bright enough to notice whether a GIF 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 GIF files 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 a baseline JPEG file to be generated. This clamps quantization values to 8 bits even at low quality settings.

−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.

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.