

**NAME**

tifftopnm – convert a TIFF file into a portable anymap

**SYNOPSIS**

**tifftopnm** [-alphaout={*alpha-filename*,-}] [-headerdump] [-respectfillorder] [*tiff-filename*]

You may abbreviate any option to its shortest unique prefix. You may use two hyphens instead of one in options. You may separate an option and its value either by an equals sign or white space.

**DESCRIPTION**

Reads a TIFF file as input. Produces a portable anymap as output. The type of the output file depends on the input file - if it's black & white, generates a *pbm* file; if it's grayscale, generates a *pgm* file; otherwise, a *ppm* file. The program tells you which type it is writing.

This program cannot read every possible TIFF file -- there are myriad variations of the TIFF format. However, it does understand monochrome and gray scale, RGB, RGBA (red/green/blue with alpha channel), CMYK (Cyan-Magenta-Yellow-Black ink color separation), and color palette TIFF files. An RGB file can have either single plane (interleaved) color or multiple plane format. The program reads 1-8 and 16 bit-per-sample input, the latter in either bigendian or littlendian encoding. Tiff directory information may also be either bigendian or littendian.

One reason this program isn't as general as TIFF programs often are is that it does not use the TIFFReadScanLine() function of the TIFF library to read TIFF files. Rather, it uses the more primitive TIFFReadScanLine() function and decodes it itself.

There is no fundamental reason that this program could not read other kinds of TIFF files; the existing limitations are mainly because no one has asked for more.

The PNM output has the same maxval as the Tiff input, except that if the Tiff input is colormapped (which implies a maxval of 65535) the PNM output has a maxval of 255. Though this may result in lost information, such input images hardly ever actually have more color resolution than a maxval of 255 provides and people often cannot deal with PNM files that have maxval > 255. By contrast, a non-colormapped Tiff image that doesn't need a maxval > 255 doesn't *have* a maxval > 255, so when we see a non-colormapped maxval > 255, we take it seriously and produce a matching output maxval.

The *tiff-filename* argument names the regular file that contains the Tiff image. If you specify "-" or don't specify this argument, **tifftopnm** uses Standard Input. In either case, the file must be seekable. That means no pipe, but any regular file is fine.

**OPTIONS**

**-alphaout=***alpha-filename*

**tifftopnm** creates a PGM (portable graymap) file containing the alpha channel values in the input image. If the input image doesn't contain an alpha channel, the *alpha-filename* file contains all zero (transparent) alpha values. If you don't specify **-alphaout**, **tifftopnm** does not generate an alpha file, and if the input image has an alpha channel, **tifftopnm** simply discards it.

If you specify - as the filename, **tifftopnm** writes the alpha output to Standard Output and discards the image.

See **pnmcomp(1)** for one way to use the alpha output file.

**-respectfillorder**

By default, **tifftopnm** ignores the "fillorder" tag in the TIFF input, which means it may incorrectly interpret the image. To make it follow the spec, use this option. For a lengthy but engaging discussion of why **tifftopnm** works this way and how to use the **-respectfillorder** option, see the note on fillorder below.

**-headerdump**

Dump TIFF file information to stderr. This information may be useful in debugging TIFF file conversion problems.

All options can be abbreviated to their shortest unique prefix.

**NOTES****Fillorder**

There is a piece of information in the header of a TIFF image called "fillorder." The TIFF specification quite clearly states that this value tells the order in which bits are arranged in a byte in the description of the image's pixels. There are two options, assuming that the image has a format where more than one pixel can be represented by a single byte: 1) the byte is filled from most significant bit to least significant bit going left to right in the image; and 2) the opposite.

However, there is confusion in the world as to the meaning of fillorder. Evidence shows that some people believe it has to do with byte order when a single value is represented by two bytes.

These people cause TIFF images to be created that, while they use a MSB-to-LSB fillorder, have a fillorder tag that says they used LSB-to-MSB. A program that properly interprets a TIFF image will not end up with the image that the author intended in this case.

For a long time, **tifftopnm** did not understand fillorder itself and assumed the fillorder was MSB-to-LSB regardless of the fillorder tag in the TIFF header. And as far as I know, there is no legitimate reason to use a fillorder other than MSB-to-LSB. So users of **tifftopnm** were happily using those TIFF images that had incorrect fillorder tags.

So that those users can continue to be happy, **tifftopnm** today continues to ignore the fillorder tag unless you tell it not to. (It does, however, warn you when the fillorder tag does not say MSB-to-LSB that the tag is being ignored).

If for some reason you have a TIFF image that actually has LSB-to-MSB fillorder, and its fillorder tag correctly indicates that, you must use the **-respectfillorder** option on **tifftopnm** to get proper results.

Examples of incorrect TIFF images are at <ftp://weather.noaa.gov>. They are apparently created by a program called **faxtotiff**.

This note was written on January 1, 2002.

**SEE ALSO**

**pnmtotiff(1)**, **pnmtotiffcmyk(1)**, **pnmcomp(1)**, **pnm(5)**

**AUTHOR**

Derived by Jef Poskanzer from `tif2ras.c`, which is Copyright (c) 1990 by Sun Microsystems, Inc. Author: Patrick J. Naughton ([naughton@wind.sun.com](mailto:naughton@wind.sun.com)).