Pamtotiff User Manual

pamtotiff

Updated: 27 March 2005
Table Of Contents

NAME

pamtotiff - convert a Netpbm image to a TIFF file

SYNOPSIS

pamtotiff [**-none** | **-packbits** | **-lzw** | **-g3** | **-g4** | **-flate** | **-adobeflate**] [**-2d**] [**-fill**] [**-predictor=***n*] [**-msb2lsb**|**-lsb2msb**] [**-rowsperstrip=***n*] [**-minisblack**|**-miniswhite**|**mb**|**mw**] [**-truecolor**] [**-color**] [**-indexbits=***bitwidthlist*]
[**-xresolution=***xres*] [**-yresolution=***yres*]
[**-resolutionunit=**{**inch** | **centimeter** | **none** | **in** | **cm** | **no**}] [**-indexbits=**[**1**[**2**[**4**[**8**]]]]] [**-append**] [**-tag=***taglist*] [*pamfile*]

You can use the minimum unique abbreviation of the options. You can use two hyphens instead of one. You can separate an option name from its value with white space instead of an equals sign.

DESCRIPTION

This program is part of Netpbm.

pamtotiff reads a PNM or PAM image as input and produces a TIFF file as output.

Actually, it handles multi-image Netpbm streams, producing multi-image TIFF streams (i.e. a TIFF stream with multiple "directories"). But before Netpbm 10.27 (March 2005), it ignored all but the first Netpbm image in the input stream.

The Output File

The output goes to Standard Output. **pamtotiff** approaches this output file differently from Unix and Netpbm convention. This is entirely due to **pamtotiff**'s use of the TIFF library to do all TIFF output.

> The output file must be seekable. **pamtotiff** does not write it sequentially. Therefore, you can't use a pipe; you can't pipe the output of **pamtotiff** to some other program. But any regular file should work.
>
> If the output file descriptor is readable, you must either specify **-append** so as to add to the existing file, or make sure the file is empty. Otherwise, **pamtotiff** will fail with an unhelpful message telling you that a TIFF library function failed to open the TIFF output stream.
>
> If you are converting multiple images (your input stream contains multiple images), the output file must be both readable and writable.

If you're using a Unix command shell to run **pamtotiff**, you use facilities of your shell to set up Standard Output. In Bash, for example, you would set up a write-only Standard Output to the file /tmp/myimage.tiff like this:

```
$ pamtotiff myimage.pnm >/tmp/myimage.tiff
```

In Bash, you would set up a read/write Standard Output to the file /tmp/myimage.tiff like this:

```
$ pamtotiff myimage.pnm 1<>/tmp/myimage.tiff
```

OPTIONS

Compression

By default, **pamtotiff** creates a TIFF file with no compression. This is your best bet most of the time. If you want to try another compression scheme or tweak some of the other even more obscure output options, there are a number of options which which to play.

Before Netpbm 8.4 (April 2000), the default was to use LZW compression. But then new releases of the TIFF library started omitting the LZW compression capability due to concern about patents on LZW. So since then, the default has been no compression. The LZW patents have now expired and new TIFF libraries do LZW, but the **pamtotiff** behavior remains the same for compatibility with older TIFF libraries and applications of **pamtotiff**.

The **-none**, **-packbits**, **-lzw**, **-g3**, **-g4**, **-flate**, and **-adobeflate** options are used to override the default and set the compression scheme used in creating the output file. The CCITT Group 3 and Group 4 compression algorithms can be used only with bilevel data. The **-2d** and **-fill** options are meaningful only with Group 3 compression: **-2d** requests 2-dimensional encoding, while **-fill** requests that each encoded scanline be zero-filled to a byte boundary. The **-predictor** option is meaningful only with LZW compression: a predictor value of 2 causes each scanline of the output image to undergo horizontal differencing before it is encoded; a value of 1 forces each scanline to be encoded without differencing. By default, **pamtotiff** creates a TIFF file with msb-to-lsb fill order. The **-msb2lsb** and **-lsb2msb** options are used to override the default and set the fill order used in creating the file.

With some older TIFF libraries, **-lzw** doesn't work because the TIFF library doesn't do LZW compression. This is because of concerns about Unisys's patent on LZW which was then in force. Actually, with very old TIFF libraries, **-lzw** works because no distributors of the TIFF library were sensitive yet to the patent issue.

-flate chooses "flate" compression, which is the patent-free compression common in the Unix world implemented by the "Z" library. It is what the PNG format uses.

Fill Order

The **-msb2lsb** and **lsb2msb** options control the fill order.

The fill order is the order in which pixels are packed into a byte in the Tiff raster, in the case that there are multiple pixels per byte. msb-to-lsb means that the leftmost columns go into the most significant bits of the byte in the Tiff image. However, there is considerable confusion about the meaning of fill order. Some believe it means whether 16 bit sample values in the Tiff image are little-endian or big-endian. This is totally erroneous (The endianness of integers in a Tiff image is designated by the image's magic number). However, ImageMagick and older Netpbm both have been known to implement that interpretation. 2001.09.06.

If the image does not have sub-byte pixels, these options have no effect other than to set the value of the FILLORDER tag in the Tiff image (which may be useful for those programs that misinterpret the tag with reference to 16 bit samples).

Color Space

-color tells **pamtotiff** to produce a color, as opposed to grayscale, TIFF image if the input is PPM, even if it contains only shades of gray. Without this option, **pamtotiff** produces a grayscale TIFF image if the input is PPM and contains only shades of gray, and at most 256 shades. Otherwise, it produces a color TIFF output. For PBM and PGM input, **pamtotiff** always produces grayscale TIFF output and this option has no effect.

The **-color** option can prevent **pamtotiff** from making two passes through the input file, thus improving speed and memory usage. See Multiple Passes.

-truecolor tells **pamtotiff** to produce the 24-bit RGB form of TIFF output if it is producing a color TIFF image. Without this option, **pamtotiff** produces a colormapped (paletted) TIFF image unless there are more than 256 colors (and in the latter case, issues a warning).

The **-truecolor** option can prevent **pamtotiff** from making two passes through the input file, thus improving speed and memory usage. See Multiple Passes.

The **-color** and **-truecolor** options did not exist before Netpbm 9.21 (December 2001).

If **pamtotiff** produces a grayscale TIFF image, this option has no effect.

The **-minisblack** and **-miniswhite** options force the output image to have a "minimum is black" or "minimum is white" photometric, respectively. If you don't specify either, **pamtotiff** uses minimum is black except when using Group 3 or Group 4 compression, in which case **pamtotiff** follows CCITT fax standards and uses "minimum is white." This usually results in better compression and is generally preferred for bilevel coding.

Before February 2001, **pamtotiff** always produced "minimum is black," due to a bug. In either case, **pamtotiff** sets the photometric interpretation tag in the TIFF output according to which photometric is actually used.

The **-indexbits** option is meaningful only for a colormapped (paletted) image. In this kind of image, the raster contains values which are indexes into a table of colors, with the indexes normally taking less space that the color description in the table. **pamtotiff** can generate indexes of 1, 2, 4, or 8 bits. By default, it will use 8, because many programs that interpret TIFF images can't handle any other width.

But if you have a small number of colors, you can make your image considerably smaller by allowing fewer than 8 bits per index, using the **-indexbits** option. The value is a comma-separated list of the bit widths you allow. **pamtotiff** chooses the smallest width you allow that allows it to index the entire color table. If you don't allow any such width, **pamtotiff** fails. Normally, the only useful value for this option is **1,2,4,8**, because a program either understands the 8 bit width (default) or understands them all.

In a Baseline TIFF image, according to the 1992 TIFF 6.0 specification, 4 and 8 are the only valid widths. There are no formal standards that allow any other values.

This option was added in June 2002. Before that, only 8 bit indices were possible.

Extra Tags

There are lots of tag types in the TIFF format that don't correspond to any information in the PNM format or to anything in the conversion process. For example, a TIFF_ARTIST tag names the artist who created the image.

You can tell **pamtotiff** explicitly to include tags such as this in its output with the **-tag** option. You identify a list of tag types and values and **pamtotiff** includes a tag in the output for each item in your list.

The value of **-tag** is the list of tags, like this example:

```
-tag=subfiletype=reducedimage,documentname=Fred,xposition=25
```

As you see, it is a list of tag specifications separated by commas. Each tag specification is a name and a value separated by an equal sign. The name is the name of the tag type, except in arbitrary upper/lower case. One place to see the names of TIFF tag types is in the TIFF library's **tiff.h** file, where there is a macro defined for each consisting of "TIFF_" plus the name. E.g. for the SUBFILETYPE tag type, there is a macro TIFF_SUBFILETYPE.

The format of the value specification for a tag (stuff after the equal sign) depends upon what kind of value the tag type has:

Integer: a decimal number

Floating point number: a decimal number

String: a string

Enumerated (For example, a 'subfiletype' tag takes an enumerated value. Its possible values are REDUCEDIMAGE, PAGE, and MASK.): The name of the value. You can see the possible value names in the TIFF library's **tiff.h** foile, where there is a macro defined for each consisting of a qualifier plus the value name. E.g. for the REDUCEDIMAGE value of a SUBFILETYPE tag, you see the macro FILETYPE_REDUCEDIMAGE.

The TIFF format assigns a unique number to each enumerated value and you can specify that number, in decimal, as an alternative. This is useful if you are using an extension of TIFF that **pamtotiff** doesn't know about.

If you specify a tag type with **-tag** that is not independent of the content of your PNM source image and **pamtotiff**'s conversion process (i.e. a tag type in which **pamtotiff** is interested), **pamtotiff** fails. For example, you cannot specify an IMAGEWIDTH tag with **-tag**, because **pamtotiff** generates an IMAGEWIDTH tag that gives the actual width of the image.

-tag was new in Netpbm 10.31 (December 2005).

Other

You can use the **-rowsperstrip** option to set the number of rows (scanlines) in each strip of data in the output file. By default, the output file has the number of rows per strip set to a value that will ensure each strip is no more than 8 kilobytes long.

The **-append** option tells **pamtotiff** to add images to the existing output file (a TIFF file may contain multiple images) instead of the default, which is to replace the output file.

-append was new in Netpbm 10.27 (March 2005).

NOTES

There are myriad variations of the TIFF format, and this program generates only a few of them. **pamtotiff** creates a grayscale TIFF file if its input is a PBM (monochrome) or PGM (grayscale) or equivalent PAM file. **pamtotiff** also creates a grayscale file if it input is PPM (color) or equivalent PAM, but there is only one color in the image.

If the input is a PPM (color) file and there are 256 colors or fewer, but more than 1, **pamtotiff** generates a color palette TIFF file. If there are more colors than that, **pamtotiff** generates an RGB (not RGBA) single plane TIFF file. Use **pnmtotiffcmyk** to generate the cyan-magenta-yellow-black ink color separation TIFF format.

The number of bits per sample in the TIFF output is determined by the maxval of the Netpbm input. If the maxval is less than 256, the bits per sample in the output is the smallest number that can encode the maxval. If the maxval is greater than or equal to 256, there are 16 bits per sample in the output.

Extra Channels

Like most Netpbm programs, **pamtotiff**'s function is mostly undefined if the input is PAM image with tuple type other than BLACKANDWHITE, GRAYSCALE, or RGB. Most of the statements in this manual assume the input is not such an exotic PAM. But there is a little defined processing of other PAM subformats.

pamtotiff assumes any 1 plane PAM image is BLACKANDWHITE or GRAYSCALE (and doesn't distinguish between those two).

pamtotiff assumes a PAM with more than 1 plane is of tuple type RGB except with that number of planes instead of 3. **pamtotiff** doesn't really understand red, green, and blue, so it has no trouble with a 2-component or 5-component color space. The TIFF format allows an arbitrary number of color compoonents, so **pamtotiff** simply maps the PAM planes directly to TIFF color components. I don't know if the meanings of 5 components in a TIFF image are standard at all, but the function is there if you want to use it.

Note that **pamtotiff** may generate either a truecolor or colormapped image with an arbitrary number of color components. In the truecolor case, the raster has that number of planes. In the colormapped case, the raster has of course 1 plane, but the color map has all the color components in it.

The most common reason for a PAM to have extra planes is when the tuple type is xxx_ALPHA, which means the highest numbered plane is a transparency plane (alpha channel). At least one user found that a TIFF with an extra plane for transparency was useful.

Note that the grayscale detection works on N-component colors, so if your planes aren't really color components, you'll want to disable this via the **-color** option.

Multiple Passes

pamtotiff reads the input image once if it can, and otherwise twice. It needs that second pass (which happens before the main pass, of course) to analyze the colors in the image and generate a color map (palette) and determine if the image is grayscale. So the second pass happens only when the input is PPM. And you can avoid it then by specifying both the **-truecolor** and **-color** options.

If the input image is small enough to fit in your system's file cache, the second pass is very fast. If not, it requires reading from disk twice, which can be slow.

When the input is from a file that cannot be rewound and reread, **pamtotiff** reads the entire input image into a temporary file which can, and works from that. Even if it needs only one pass.

Before Netpbm 9.21 (December 2001), **pamtotiff** always read the entire image into virtual memory and then did one, two, or three passes through the memory copy. The **-truecolor** and **-color** options did not exist. The passes through memory would involve page faults if the entire image did not fit into real memory. The image in memory required considerably more memory (12 bytes per pixel) than the cached file version of the image would.

Resolution

A Tiff image may contain information about the resolution of the image, which means how big in real dimensions (centimeters, etc.) each pixel in the raster is. That information is in the TIFF XRESOLUTION, YRESOLUTION, and RESOLUTIONUNIT tags. By default, **pamtotiff** does not include any tags of these types, but you can specify them with the **-tags** option.

There are also options **-xresolution**, **-yresolution**, and **-resolutionunit**, but those are obsolete. Before **-tags** existed (before Netpbm 10.31 (December 2005), they were the only way.

Note that the number of pixels in the image and how much information each contains is determined independently from the setting of the resolution tags. The number of pixels in the output is the same as in the input, and each pixel contains the same information. For your resolution tags to be meaningful, they have to consistent with whatever created the PNM input. E.g. if a scanner turned a 10 centimeter wide image into a 1000 pixel wide PNM image, then your horizontal resolution is 100 pixels per centimeter, and if your XRESOLUTION tag says anything else, something that prints your TIFF image won't print the proper 10 centimeter image.

The value of the XRESOLUTION tag is a floating point decimal number that tells how many pixels there are per unit of distance in the horizontal direction. **-yresolution** is analogous for the vertical direction.

The unit of distance is given by the value of the RESOLUTIONUNIT option. That value is either INCH, CENTIMETER, or NONE. NONE means the unit is arbitrary or unspecified. This could mean that the creator and user of the image have a separate agreement as to what the unit is. But usually, it just means that the horizontal and vertical resolution values cannot be used for anything except to determine aspect ratio (because even though the unit is arbitrary or unspecified, it has to be the same for both resolution numbers).

If you don't use a **-tag** option to specify the resolution tag and use the obsolete options instead, note the following:

> If you don't include an specify **-xresolution**, the Tiff image does not contain horizontal resolution information. Likewise for **-yresolution**. If you don't specify **-resolutionunit**, the default is inches.
>
> Before Netpbm 10.16 (June 2003), **-resolutionunit** did not exist and the resolution unit was always

inches.

HISTORY

pamtotiff was originally **pnmtotiff** and did not handle PAM input. It was extended and renamed in Netpbm 10.30 (October 2005).

SEE ALSO

tifftopnm, **pnmtotiffcmyk**, **pamdepth**, **pamtopnm**, **pam**

AUTHOR

Derived by Jef Poskanzer from ras2tiff.c, which is Copyright (c) 1990 by Sun Microsystems, Inc. Author: Patrick J. Naughton (naughton@wind.sun.com).

Table Of Contents