NAME

pbmtojbg - portable bitmap to JBIG1 file converter

SYNOPSIS

pbmtojbg [*options*] [*input-file* | **-** [*output-file*]]

DESCRIPTION

Reads in a portable bitmap (PBM) from a file or standard input, compresses it, and outputs the image as a *JBIG1* bi-level image entity (BIE) file.

JBIG1 is a highly effective lossless compression algorithm for bi-level images (one bit per pixel), which is particularly suitable for scanned document pages.

A *JBIG1* encoded image can be stored in several resolutions (progressive mode). These resolution layers can be stored all in one single BIE or they can be stored in several separate BIE files. All resolution layers except the lowest one are stored merely as differences to the next lower resolution layer, because this requires less space than encoding the full image completely every time. Each resolution layer has twice the number of horizontal and vertical pixels than the next lower layer. *JBIG1* files can also store several bits per pixel as separate bitmap planes, and *pbmtojbg* can read a PGM file and transform it into a multi-bitplane BIE.

OPTIONS

- A single hyphen instead of an input file name will cause *pbmtojbg* to read the data from standard input instead from a file.

-q Encode the image in one single resolution layer (sequential mode). This is usually the most efficient compression method. By default, the number of resolution layers is chosen automatically such that the lowest layer image is not larger than 640 x 480 pixels. This is a shortcut for **-d 0**.

-x *number* Specify the maximal horizontal size of the lowest resolution layer. The default is 640 pixels.

-y *number* Specify the maximal vertical size of the lowest resolution layer. The default is 480 pixels.

-l *number* Select the lowest resolution layer that will be written to the BIE. It is possible to store the various resolution layers of a *JBIG1* image in progressive mode into different BIEs. Options **-l** and **-h** allow to select the resolution-layer interval that will appear in

the created BIE. The lowest resolution layer has number 0 and this is also the default value. By default all layers will be written.

-h *number* Select the highest resolution layer that will be written to the BIE. By default all layers will be written. See also option **-l**.

-b Use binary values instead of Gray code words in order to encode pixel values in multiple bitplanes. This option has only an effect if the input is a PGM file and if more than one bitplane is produced. Note that the decoder has to make the same selection but cannot determine from the BIE, whether Gray or binary code words were used by the encoder.

-d *number* Specify the total number of differential resolution layers into which the input image will be split in addition to the lowest layer. Each additional layer reduces the size of layer 0 by 50 %. This option overrides options **-x** and **-y** which are usually a more comfortable way of selecting the number of resolution layers.

-s *number* The *JBIG1* algorithm splits each image into a number of horizontal stripes. This option specifies that each stripe shall have *number* lines in layer 0. The default value is selected so that approximately 35 stripes will be used for the whole image.

-m *number* Select the maximum horizontal offset of the adaptive template pixel. The *JBIG1* encoder uses ten neighbour pixels to estimate the probability of the next pixel being black or white. It can move one out of these ten pixels. This is especially useful for dithered images, as long as the distance of this adaptive pixel can be adjusted to the period of the dither pattern. By default, the adaptive template pixel is allowed to move up to 8 pixels away horizontally. This encoder supports distances up to 127 pixels. Annex A of the standard suggests that decoders should support at least a horizontal distance of 16 pixels, so using values not higher than 16 for *number* might increase the chances of interoperability with other *JBIG1* implementations. On the other hand, the T.85 fax application profile requires decoders to support horizontal offsets up to 127 pixels, which the maximum value permitted by the standard. (The maximal vertical offset of the adaptive template pixel is always zero for this encoder.)

-t *number* Encode only the specified number of most significant bit planes. This option allows to reduce the depth of an input PGM file if not all bits per pixel are needed in the output.

-o *number* *JBIG1* separates an image into several horizontal stripes, resolution layers and planes, were each plane contains one bit per pixel. One single stripe in one plane and layer is encoded as a data unit called stripe data entity (SDE) inside the BIE. There are 12

different possible orders in which the SDEs can be stored inside the BIE and *number* selects which one shall be used. The order of the SDEs is only relevant for applications that want to decode a *JBIG1* file which has not yet completely arrived from e.g. a slow network connection. For instance some applications prefer that the outermost of the three loops (stripes, layers, planes) is over all layers so that all data of the lowest resolution layer is transmitted first.
The following values for *number* select these loop arrangements for writing the SDEs (outermost loop first):

0 planes, layers, stripes
2 layers, planes, stripes
3 layers, stripes, planes
4 stripes, planes, layers
5 planes, stripes, layers
6 stripes, layers, planes

All loops count starting with zero, however by adding 8 to the above order code, the layer loop can be reversed so that it counts down to zero and then higher resolution layers will be stored before lower layers. Default order is 3 which writes at first all planes of the first stripe and then completes layer 0 before continuing with the next layer and so on.

-p *number* This option allows to activate or deactivate various optional algorithms defined in the *JBIG1* standard. Just add the numbers of the following options which you want to activate in order to get the *number* value:

4 deterministic prediction (DPON)
8 layer 0 typical prediction (TPBON)
16 diff. layer typ. pred. (TPDON)
64 layer 0 two-line template (LRLTWO)

Except for special applications (like communication with *JBIG1* subset implementations) and for debugging purposes you will normally not want to change anything here. The default is 28, which provides the best compression result.

-C *string* Add the *string* in a comment marker segment to the produced data stream. (There is no support at present for adding comments that contain the zero byte.)

-c Determine the adaptive template pixel movement as suggested in annex C of the standard. By default the template change takes place directly in the next line, which is

most effective. However, a few conformance test examples in the standard require the adaptive template change to be delayed until the first line of the next stripe. This option selects this special behavior, which is normally not required except in order to pass some conformance tests.

-r Use the SDRST marker instead of the normal SDNORM marker. The probably only useful application of this option is to generate test data for checking whether a *JBIG1* decoder has implemented SDRST correctly. In a normal *JBIG1* data stream, each stripe data entity (SDE) is terminated by an SDNORM marker, which preserves the state of the arithmetic encoder (and more) for the next stripe in the same layer. The alternative SDRST marker resets this state at the end of the stripe.

-Y *number* A long time ago, there were fax machines that couldn't even hold a single page in memory. They had to start transmitting data before the page was scanned in completely and the length of the image was known. The authors of the standard added a rather ugly hack to the otherwise beautiful JBIG1 format to support this. The NEWLEN marker segment can override the image height stated in the BIE header anywhere later in the data stream. Normally *pbmtojbg* never generates NEWLEN marker segments, as it knows the correct image height when it outputs the header. This option is solely intended for the purpose of generating test files with NEWLEN marker segments. It can be used to specify a higher initial image height for use in the BIE header, and *pbmtojbg* will then add a NEWLEN marker segment at the latest possible opportunity to the data stream to signal the correct final height.

-f This option makes the output file comply to the "facsimile application profile" defined in ITU-T Recommendation T.85. It is a shortcut for **-q -o 0 -p 8 -s 128 -t 1 -m 127**.

-v After the BIE has been created, a few technical details of the created file will be listed (verbose mode).

BUGS

Using standard input and standard output for binary data works only on systems where there is no difference between binary and text streams (e.g., Unix). On other systems (e.g., MS-DOS), using standard input or standard output may cause control characters like CR or LF to be inserted or deleted and this will damage the binary data.

STANDARDS

This program implements the *JBIG1* image coding algorithm as specified in ISO/IEC 11544:1993 and ITU-T T.82(1993).

AUTHOR

The *pbmtojbg* program is part of the *JBIG-KIT* package, which has been developed by Markus Kuhn. The most recent version of this portable *JBIG1* library and tools set is available from <http://www.cl.cam.ac.uk/~mgk25/jbigkit/>.

SEE ALSO

pbm(5), pgm(5), jbgtopbm(1)