

Epeg Reference Manual

Generated by Doxygen 1.5.1

Sat Aug 16 00:06:10 2008

Contents

1 Epeg Library Documentation	1
1.1 What is Epeg?	1
2 Epeg File Index	3
2.1 Epeg File List	3
3 Epeg Page Index	5
3.1 Epeg Related Pages	5
4 Epeg File Documentation	7
4.1 epeg.c File Reference	7
5 Epeg Page Documentation	15
5.1 Todo List	15

Chapter 1

Epeg Library Documentation

Version:

0.9.0

Author:

Carsten Haitzler <raster@rasterman.com>

Date:

2000-2003

1.1 What is Epeg?

An IMMENSELY FAST JPEG thumbnailer library API.

Why write this? It's a convenience library API to using libjpeg to load JPEG images destined to be turned into thumbnails of the original, saving information with these thumbnails, retrieving it and managing to load the image ready for scaling with the minimum of fuss and CPU overhead.

This means it's insanely fast at loading large JPEG images and scaling them down to tiny thumbnails. It's speedup will be proportional to the size difference between the source image and the output thumbnail size as a count of their pixels.

It makes use of libjpeg features of being able to load an image by only decoding the DCT coefficients needed to reconstruct an image of the size desired. This gives a massive speedup. If you do not try and access the pixels in a format other than YUV (or GRAY8 if the source is grayscale) then it also avoids colorspace conversions as well.

Using the library is very easy, look at this example:

```
#include "Epeg.h"

int
main(int argc, char **argv)
{
    Epeg_Image *im;
```

```
if (argc != 3)
{
    printf("Usage: %s input.jpg thumb.jpg\n", argv[0]);
    exit(0);
}
im = epeg_file_open(argv[1]);
if (!im)
{
    printf("Cannot open %s\n", argv[1]);
    exit(-1);
}

epeg_decode_size_set      (im, 128, 96);
epeg_quality_set         (im, 75);
epeg_thumbnail_comments_enable (im, 1);
epeg_file_output_set     (im, argv[2]);
epeg_encode              (im);
epeg_close               (im);

return 0;
}
```

You can compile this program with as small a line as:

```
gcc epeg_test.c -o epeg_test `epeg-config --cflags --libs`
```

It is a very simple library that just makes life easier when trying to generate lots of thumbnails for large JPEG images as quickly as possible. Your milage may vary, but it should save you lots of time and effort in using libjpeg in general.

Todo

Check all input parameters for sanity.

Todo

Actually report error conditions properly.

Chapter 2

Epeg File Index

2.1 Epeg File List

Here is a list of all documented files with brief descriptions:

[epeg.c](#) (Epeg JPEG Thumbnailer library) 7

Chapter 3

Epeg Page Index

3.1 Epeg Related Pages

Here is a list of all related documentation pages:

Todo List	15
-------------------------------------	----

Chapter 4

Epeg File Documentation

4.1 epeg.c File Reference

Epeg JPEG Thumbnailer library.

Functions

- EAPI Epeg_Image * [epeg_file_open](#) (const char *file)
Open a JPEG image by filename.
- EAPI EAPI Epeg_Image * [epeg_memory_open](#) (unsigned char *data, int size)
Open a JPEG image stored in memory.
- EAPI void [epeg_size_get](#) (Epeg_Image *im, int *w, int *h)
Return the original JPEG pixel size.
- EAPI void [epeg_colorspace_get](#) (Epeg_Image *im, int *space)
Return the original JPEG pixel color space.
- EAPI void [epeg_decode_size_set](#) (Epeg_Image *im, int w, int h)
Set the size of the image to decode in pixels.
- EAPI void [epeg_decode_colorspace_set](#) (Epeg_Image *im, Epeg_Colorspace colorspace)
Set the colorspace in which to decode the image.
- EAPI const void * [epeg_pixels_get](#) (Epeg_Image *im, int x, int y, int w, int h)
Get a segment of decoded pixels from an image.
- EAPI const void * [epeg_pixels_get_as_RGB8](#) (Epeg_Image *im, int x, int y, int w, int h)
Get a segment of decoded pixels from an image.

- EAPI void `jpeg_pixels_free` (`Jpeg_Image *im`, `const void *data`)
Free requested pixel block from an image.
- EAPI `const char *` `jpeg_comment_get` (`Jpeg_Image *im`)
Get the image comment field as a string.
- EAPI void `jpeg_thumbnail_comments_get` (`Jpeg_Image *im`, `Jpeg_Thumbnail_Info *info`)
Get thumbnail comments of loaded image.
- EAPI void `jpeg_comment_set` (`Jpeg_Image *im`, `const char *comment`)
Set the comment field of the image for saving.
- EAPI void `jpeg_quality_set` (`Jpeg_Image *im`, `int quality`)
Set the encoding quality of the saved image.
- EAPI void `jpeg_thumbnail_comments_enable` (`Jpeg_Image *im`, `int onoff`)
Enable thumbnail comments in saved image.
- EAPI void `jpeg_file_output_set` (`Jpeg_Image *im`, `const char *file`)
Set the output file path for the image when saved.
- EAPI void `jpeg_memory_output_set` (`Jpeg_Image *im`, `unsigned char **data`, `int *size`)
Set the output file to be a block of allocated memory.
- EAPI `int` `jpeg_encode` (`Jpeg_Image *im`)
This saves the image to its specified destination.
- EAPI `int` `jpeg_trim` (`Jpeg_Image *im`)
FIXME: Document this.
- EAPI void `jpeg_close` (`Jpeg_Image *im`)
Close an image handle.

4.1.1 Detailed Description

Epeg JPEG Thumbnailer library.

These routines are used for the Epeg library.

4.1.2 Function Documentation

4.1.2.1 EAPI void `jpeg_close` (`Jpeg_Image * im`)

Close an image handle.

Parameters:

im A handle to an opened Epeg image.

This closes an opened image handle and frees all memory associated with it. It does not free encoded data generated by `jpeg_memory_output_set()` followed by `jpeg_encode()` nor does it guarantee to free any data received by `jpeg_pixels_get()`. Once an image handle is closed consider it invalid.

4.1.2.2 EAPI void jpeg_colorspace_get (Epeg_Image * im, int * space)

Return the original JPEG pixel color space.

Parameters:

im A handle to an opened Epeg image.

space A pointer to the color space value to be filled in.

Returns the image color space.

4.1.2.3 EAPI const char* jpeg_comment_get (Epeg_Image * im)

Get the image comment field as a string.

Parameters:

im A handle to an opened Epeg image.

Returns:

A pointer to the loaded image comments.

This function returns the comment field as a string (NUL byte terminated) of the loaded image *im*, if there is a comment, or NULL if no comment is saved with the image. Consider the string returned to be read-only.

4.1.2.4 EAPI void jpeg_comment_set (Epeg_Image * im, const char * comment)

Set the comment field of the image for saving.

Parameters:

im A handle to an opened Epeg image.

comment The comment to set.

Set the comment for the image file for when it gets saved. This is a NUL byte terminated C string. If *comment* is NULL the output file will have no comment field.

The default comment will be any comment loaded from the input file.

4.1.2.5 EAPI void epeg_decode_colorspace_set (Epeg_Image * *im*, Epeg_Colorspace *colorspace*)

Set the colorspace in which to decode the image.

Parameters:

- im* A handle to an opened Epeg image.
- colorspace* The colorspace to decode the image in.

This sets the colorspace to decode the image in. The default is EPEG_YUV8, as this is normally the native colorspace of a JPEG file, avoiding any colorspace conversions for a faster load and/or save.

4.1.2.6 EAPI void epeg_decode_size_set (Epeg_Image * *im*, int *w*, int *h*)

Set the size of the image to decode in pixels.

Parameters:

- im* A handle to an opened Epeg image.
- w* The width of the image to decode at, in pixels.
- h* The height of the image to decode at, in pixels.

Sets the size at which to decode the JPEG image, giving an optimised load that only decodes the pixels needed.

4.1.2.7 EAPI int epeg_encode (Epeg_Image * *im*)

This saves the image to its specified destination.

Parameters:

- im* A handle to an opened Epeg image.

This saves the image *im* to its destination specified by [epeg_file_output_set\(\)](#) or [epeg_memory_output_set\(\)](#). The image will be encoded at the decoded pixel size, using the quality, comment and thumbnail comment settings set on the image.

retval 1 - error scale 2 - error encode 3 - error decode 4 - error decode (setjmp)

4.1.2.8 EAPI Epeg_Image* epeg_file_open (const char * *file*)

Open a JPEG image by filename.

Parameters:

- file* The file path to open.

Returns:

- A handle to the opened JPEG file, with the header decoded.

This function opens the file indicated by the `file` parameter, and attempts to decode it as a jpeg file. If this fails, NULL is returned. Otherwise a valid handle to an open JPEG file is returned that can be used by other Epeg calls.

The `file` must be a pointer to a valid C string, NUL (0 byte) terminated that is a relative or absolute file path. If not results are not determined.

See also: [epeg_memory_open\(\)](#), [epeg_close\(\)](#)

4.1.2.9 EAPI void epeg_file_output_set (Epeg_Image * *im*, const char * *file*)

Set the output file path for the image when saved.

Parameters:

im A handle to an opened Epeg image.

file The path to the output file.

This sets the output file path name (either a full or relative path name) to where the file will be written when saved. `file` must be a NUL terminated C string containing the path to the file to be saved to. If it is NULL, the image will not be saved to a file when calling [epeg_encode\(\)](#).

4.1.2.10 EAPI EAPI Epeg_Image* epeg_memory_open (unsigned char * *data*, int *size*)

Open a JPEG image stored in memory.

Parameters:

data A pointer to the memory containing the JPEG data.

size The size of the memory segment containing the JPEG.

Returns:

A handle to the opened JPEG, with the header decoded.

This function opens a JPEG file that is stored in memory pointed to by `data`, and that is `size` bytes in size. If successful a valid handle is returned, or on failure NULL is returned.

See also: [epeg_file_open\(\)](#), [epeg_close\(\)](#)

4.1.2.11 EAPI void epeg_memory_output_set (Epeg_Image * *im*, unsigned char ** *data*, int * *size*)

Set the output file to be a block of allocated memory.

Parameters:

im A handle to an opened Epeg image.

data A pointer to a pointer to a memory block.

size A pointer to a counter of the size of the memory block.

This sets the output encoding of the image when saved to be allocated memory. After `jpeg_close()` is called the pointer pointed to by `data` and the integer pointed to by `size` will contain the pointer to the memory block and its size in bytes, respectively. The memory block can be freed with the `free()` function call. If the save fails the pointer to the memory block will be unaffected, as will the size.

4.1.2.12 EAPI void jpeg_pixels_free (Epeg_Image * *im*, const void * *data*)

Free requested pixel block from an image.

Parameters:

- im* A handle to an opened Epeg image.
- data* The pointer to the image pixels.

This frees the data for a block of pixels requested from image `im`. `data` must be a valid (non NULL) pointer to a pixel block taken from the image `im` by `jpeg_pixels_get()` and must be called before the image is closed by `jpeg_close()`.

4.1.2.13 EAPI const void* jpeg_pixels_get (Epeg_Image * *im*, int *x*, int *y*, int *w*, int *h*)

Get a segment of decoded pixels from an image.

Parameters:

- im* A handle to an opened Epeg image.
- x* Rectangle X.
- y* Rectangle Y.
- w* Rectangle width.
- h* Rectangle height.

Returns:

Pointer to the top left of the requested pixel block.

Return image pixels in the decoded format from the specified location rectangle bounded with the box `x`, `y` `w` X `y`. The pixel block is packed with no row padding, and it organized from top-left to bottom right, row by row. You must free the pixel block using `jpeg_pixels_free()` before you close the image handle, and assume the pixels to be read-only memory.

On success the pointer is returned, on failure, NULL is returned. Failure may be because the rectangle is out of the bounds of the image, memory allocations failed or the image data cannot be decoded.

4.1.2.14 EAPI const void* jpeg_pixels_get_as_RGB8 (Epeg_Image * *im*, int *x*, int *y*, int *w*, int *h*)

Get a segment of decoded pixels from an image.

Parameters:

- im* A handle to an opened Epeg image.

x Rectangle X.
y Rectangle Y.
w Rectangle width.
h Rectangle height.

Returns:

Pointer to the top left of the requested pixel block.

Return image pixels in the decoded format from the specified location rectangle bounded with the box *x*, *y* *w* X *y*. The pixel block is packed with no row padding, and it organised from top-left to bottom right, row by row. You must free the pixel block using `epeg_pixels_free()` before you close the image handle, and assume the pixels to be read-only memory.

On success the pointer is returned, on failure, NULL is returned. Failure may be because the rectangle is out of the bounds of the image, memory allocations failed or the image data cannot be decoded.

4.1.2.15 EAPI void epeg_quality_set (Epeg_Image * *im*, int *quality*)

Set the encoding quality of the saved image.

Parameters:

im A handle to an opened Epeg image.
quality The quality of encoding from 0 to 100.

Set the quality of the output encoded image. Values from 0 to 100 inclusive are valid, with 100 being the maximum quality, and 0 being the minimum. If the quality is set equal to or above 90%, the output U and V color planes are encoded at 1:1 with the Y plane.

The default quality is 75.

4.1.2.16 EAPI void epeg_size_get (Epeg_Image * *im*, int * *w*, int * *h*)

Return the original JPEG pixel size.

Parameters:

im A handle to an opened Epeg image.
w A pointer to the width value in pixels to be filled in.
h A pointer to the height value in pixels to be filled in.

Returns the image size in pixels.

4.1.2.17 EAPI void epeg_thumbnail_comments_enable (Epeg_Image * *im*, int *onoff*)

Enable thumbnail comments in saved image.

Parameters:

im A handle to an opened Epeg image.

onoff A boolean on and off enabling flag.

if *onoff* is 1, the output file will have thumbnail comments added to it, and if it is 0, it will not. The default is 0.

**4.1.2.18 EAPI void epeg_thumbnail_comments_get (Epeg_Image * *im*,
Epeg_Thumbnail_Info * *info*)**

Get thumbnail comments of loaded image.

Parameters:

im A handle to an opened Epeg image.

info Pointer to a thumbnail info struct to be filled in.

This function retrieves thumbnail comments written by Epeg to any saved JPEG files. If no thumbnail comments were saved, the fields will be 0 in the *info* struct on return.

4.1.2.19 EAPI int epeg_trim (Epeg_Image * *im*)

FIXME: Document this.

Parameters:

im A handle to an opened Epeg image.

FIXME: Document this.

Chapter 5

Epeg Page Documentation

5.1 Todo List

page [Epeg Library Documentation](#) Check all input parameters for sanity.

page [Epeg Library Documentation](#) Actually report error conditions properly.

Index

epg.c, [7](#)
 epg_close, [8](#)
 epg_colorspace_get, [9](#)
 epg_comment_get, [9](#)
 epg_comment_set, [9](#)
 epg_decode_colorspace_set, [9](#)
 epg_decode_size_set, [10](#)
 epg_encode, [10](#)
 epg_file_open, [10](#)
 epg_file_output_set, [11](#)
 epg_memory_open, [11](#)
 epg_memory_output_set, [11](#)
 epg_pixels_free, [12](#)
 epg_pixels_get, [12](#)
 epg_pixels_get_as_RGB8, [12](#)
 epg_quality_set, [13](#)
 epg_size_get, [13](#)
 epg_thumbnail_comments_enable, [13](#)
 epg_thumbnail_comments_get, [14](#)
 epg_trim, [14](#)
epg_close
 epg.c, [8](#)
epg_colorspace_get
 epg.c, [9](#)
epg_comment_get
 epg.c, [9](#)
epg_comment_set
 epg.c, [9](#)
epg_decode_colorspace_set
 epg.c, [9](#)
epg_decode_size_set
 epg.c, [10](#)
epg_encode
 epg.c, [10](#)
epg_file_open
 epg.c, [10](#)
epg_file_output_set
 epg.c, [11](#)
epg_memory_open
 epg.c, [11](#)
epg_memory_output_set
 epg.c, [11](#)
epg_pixels_free
 epg.c, [12](#)
epg_pixels_get
 epg.c, [12](#)
epg_pixels_get_as_RGB8
 epg.c, [12](#)
epg_quality_set
 epg.c, [13](#)
epg_size_get
 epg.c, [13](#)
epg_thumbnail_comments_enable
 epg.c, [13](#)
epg_thumbnail_comments_get
 epg.c, [14](#)
epg_trim
 epg.c, [14](#)