0.1 epeg.c File Reference

Epeg JPEG Thumbnailer library.

Functions

- Epeg_Image ∗ epeg_file_open (const char ∗file)

  Open a JPEG image by filename.

- Epeg_Image ∗ epeg_memory_open (unsigned char ∗data, int size)

  Open a JPEG image stored in memory.

- void epeg_size_get (Epeg_Image ∗im, int ∗w, int ∗h)

  Return the original JPEG pixel size.

- void epeg_decode_size_set (Epeg_Image ∗im, int w, int h)

  Set the size of the image to decode in pixels.

- void epeg_decode_colorspace_set (Epeg_Image ∗im, Epeg_Colorspace colorspace)

  Set the colorspace in which to decode the image.

- 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.

- void epeg_pixels_free (Epeg_Image ∗im, const void ∗data)

  Free requested pixel block from an image.

- const char ∗ epeg_comment_get (Epeg_Image ∗im)

  Get the image comment field as a string.

- void epeg_thumbnail_comments_get (Epeg_Image ∗im, Epeg_Thumbnail_Info ∗info)

  Get thumbnail comments of loaded image.

- void epeg_comment_set (Epeg_Image ∗im, const char ∗comment)

  Set the comment field of the image for saving.

- void epeg_quality_set (Epeg_Image ∗im, int quality)

  Set the encoding quality of the saved image.

- void epeg_thumbnail_comments_enable (Epeg_Image ∗im, int onoff)

  Enable thumbnail comments in saved image.

- void epeg_file_output_set (Epeg_Image ∗im, const char ∗file)

  Set the output file path for the image when saved.

- void epeg_memory_output_set (Epeg_Image ∗im, unsigned char ∗∗data, int ∗size)

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

- void epeg_encode (Epeg_Image ∗im)

  This saved the image to its specified destination.

- void epeg_close (Epeg_Image ∗im)

  Close an image handle.

0.1.1 Detailed Description

Epeg JPEG Thumbnailer library.

These routines are used for the Epeg library.

0.1.2 Function Documentation

0.1.2.1 void epeg_close (Epeg_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 epeg_memory_output_set() followed by epeg_encode() nor does it guarantee to free any data recieved by epeg_pixels_get(). Once an image handle is closed consider it invalid.

0.1.2.2 const char$*$ epeg_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.

0.1.2.3 void epeg_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.

0.1.2.4 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.

0.1.2.5 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 deocode the JPEG image, giving an optimised load that only decodes the pixels needed.

0.1.2.6 void epeg_encode (Epeg_Image ∗ *im*)

This saved 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 deoded pixel size, using the quality, comment and thumbnail comment settings set on the image.

0.1.2.7 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 failes, 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 thats is a relative or absolute file path. If not results are not determined.

See also: epeg_memory_open(), epeg_close()

0.1.2.8 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 conatining 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().

0.1.2.9 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()

0.1.2.10 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 epeg_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, respecitvely. 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.

0.1.2.11 void epeg_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 epeg_pixels_get() and mustbe called before the image is closed by epeg_close().

0.1.2.12 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.

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 organsied 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.

0.1.2.13 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.

0.1.2.14 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.

0.1.2.15 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.

0.1.2.16 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.