

mylib

Generated by Doxygen 1.9.1

1 README	1
2 Class Index	3
2.1 Class List	3
3 File Index	5
3.1 File List	5
4 Class Documentation	7
4.1 bmp_image_t Struct Reference	7
4.2 surround_window_t Struct Reference	7
5 File Documentation	9
5.1 libsurround.h File Reference	9
5.1.1 Function Documentation	10
5.1.1.1 bmp_image_get_height()	10
5.1.1.2 bmp_image_get_pixel()	11
5.1.1.3 bmp_image_get_width()	11
5.1.1.4 create_bmp()	12
5.1.1.5 err_exit()	12
5.1.1.6 err_exit_alt()	12
5.1.1.7 err_msg()	13
5.1.1.8 filename_from_pathname()	13
5.1.1.9 set_pixel()	13
5.1.1.10 usage_err()	15
Index	17

Chapter 1

README

this is a readme file

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

bmp_image_t	7
surround_window_t	7

Chapter 3

File Index

3.1 File List

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

libsurround.h	9
---	---

Chapter 4

Class Documentation

4.1 bmp_image_t Struct Reference

Public Attributes

- FILE * **fptr**

The documentation for this struct was generated from the following file:

- [libsurround.h](#)

4.2 surround_window_t Struct Reference

Public Attributes

- xcb_window_t * **window**
- xcb_screen_t * **screen**
- xcb_connection_t * **surround_connection**

The documentation for this struct was generated from the following file:

- [libsurround.h](#)

Chapter 5

File Documentation

5.1 libsurround.h File Reference

```
#include <xcb/xcb.h>
#include <stddef.h>
#include <stdint.h>
#include <stdio.h>
#include <unistd.h>
#include <sys/stat.h>
#include <fcntl.h>
#include "ename.c.inc"
```

Classes

- struct [surround_window_t](#)
- struct [bmp_image_t](#)

Macros

- #define **u8** uint8_t
- #define **u16** uint16_t
- #define **u32** uint32_t
- #define **u64** uint64_t
- #define **i8** int8_t
- #define **i16** int16_t
- #define **i32** int32_t
- #define **i64** int64_t
- #define **curr**(fd) lseek(fd,0,SEEK_CUR);

Typedefs

- typedef int **sur_file_t**
- typedef uint32_t **rgb_color_t**

Enumerations

- enum `bool` { `false` , `true` }
bool enum type; defines false as 0 and true as 1

Functions

- int `create_window` (`surround_window_t` *win)
- int `free_window` ()
- int `drawer_init` ()
- int `drawer_exit` ()
- int `map_window` ()
- int `unmap_window` ()
- void `err_exit` (const char *format,...)
prints an error message and exits with EXIT_FAILURE
- void `err_exit_alt` (const char *format,...)
prints an error message and exits with EXIT_FAILURE without calling exit handlers
- void `err_msg` (const char *format,...)
prints an error message
- void `usage_err` (const char *format,...)
used to inform user of error in command-line argument usage
- char * `filename_from_pathname` (const char *pathname, int len)
gets a filename from a pathname allocates a string containing the part of the string in pathname after all the '/' characters ex: given pathname "dir1/dir2/dir3/file1" will return "file1" TODO: make it work with '\'
- int `bmp_image_get_width` (`bmp_image_t` image)
get width of image
- int `bmp_image_get_height` (`bmp_image_t` image)
get height of image
- rgb_color_t `bmp_image_get_pixel` (uint32_t x, uint32_t y, `bmp_image_t` image)
get color in rgb of given pixel of given image
- int `set_pixel` (uint32_t x, uint32_t y, rgb_color_t color, `bmp_image_t` image)
set a pixel to a given color in a given image
- `bmp_image_t` `create_bmp` (const char *filename, uint32_t width, uint32_t height, uint32_t xres, uint32_t yres)
create and return a bmp file
- unsigned int `get_pixel` (unsigned int x, unsigned int y, `bmp_image_t` image)
- void `set_pixel` (unsigned int x, unsigned int y, unsigned int color, `bmp_image_t` image)
- `bmp_image_t` `create_bmp` (unsigned int width, unsigned int height, unsigned int xres, unsigned int yres)

5.1.1 Function Documentation

5.1.1.1 `bmp_image_get_height()`

```
unsigned int bmp_image_get_height (
    bmp_image_t image )
```

get height of image

Parameters

<i>image</i>	image to get height of
--------------	------------------------

Returns

height on success, -1 on failure with errno set accordingly

5.1.1.2 bmp_image_get_pixel()

```
rgb_color_t bmp_image_get_pixel (
    uint32_t x,
    uint32_t y,
    bmp_image_t image )
```

get color in rgb of given pixel of given image

Parameters

<i>x</i>	x-coordinate of pixel
<i>y</i>	y-coordinate of pixel
<i>image</i>	image to get pixel from

Returns

rgb_color_t with first byte as red value, second as green, third as blue, -1 on failure with errno set accordingly

5.1.1.3 bmp_image_get_width()

```
unsigned int bmp_image_get_width (
    bmp_image_t image )
```

get width of image

Parameters

<i>image</i>	image to get width of
--------------	-----------------------

Returns

width on success, -1 on failure with errno set accordingly

5.1.1.4 create_bmp()

```
bmp_image_t create_bmp (
    const char * filename,
    uint32_t width,
    uint32_t height,
    uint32_t xres,
    uint32_t yres )
```

create and return a bmp file

Parameters

<i>filename</i>	filename of bmp file to be created
<i>width</i>	width of image to be stored in bmp
<i>height</i>	height of image to be stored in bmp
<i>xres</i>	x resolution of image to be stored in bmp
<i>yres</i>	y resolution of image to be stored in bmp

Returns

file descriptor of image file or -1 on failure with errno set accordingly

5.1.1.5 err_exit()

```
void err_exit (
    const char * format,
    ... )
```

prints an error message and exits with EXIT_FAILURE

Parameters

<i>format</i>	a format string to be printed(no newline required)
...	values to be formatted into output string

5.1.1.6 err_exit_alt()

```
void err_exit_alt (
    const char * format,
    ... )
```

prints an error message and exits with EXIT_FAILURE without calling exit handlers

Parameters

<i>format</i>	a format string to be printed(no newline required)
...	values to be formatted into output string

5.1.1.7 err_msg()

```
void err_msg (
    const char * format,
    ... )
```

prints an error message

Parameters

<i>format</i>	a format string to be printed(no newline required)
...	values to be formatted into output string

5.1.1.8 filename_from_pathname()

```
char * filename_from_pathname (
    const char * pathname,
    int len )
```

gets a filename from a pathname allocates a string containing the part of the string in pathname after all the '/' characters ex: given pathname "dir1/dir2/dir3/file1" will return "file1" TODO: make it work with '\'

Parameters

<i>pathname</i>	pathname to get filename from
<i>len</i>	length of param pathname; for most use cases just set this to strlen(pathname)

Returns

a dynamically allocated string containing the filename

5.1.1.9 set_pixel()

```
int set_pixel (
    uint32_t x,
    uint32_t y,
```

```
    rgb_color_t color,  
    bmp_image_t image )
```

set a pixel to a given color in a given image

Parameters

<i>x</i>	x-coordinate of pixel
<i>y</i>	y-coordinate of pixel
<i>color</i>	color to set pixel to
<i>image</i>	image to set pixel of

Returns

-1 on failure with errno set accordingly

5.1.1.10 usage_err()

```
void usage_err (
    const char * format,
    ... )
```

used to inform user of error in command-line argument usage

Parameters

<i>format</i>	a format string to be printed(newline required)
...	values to be formatted into the output string

Index

- bmp_image_get_height
 - libsurround.h, [10](#)
- bmp_image_get_pixel
 - libsurround.h, [11](#)
- bmp_image_get_width
 - libsurround.h, [11](#)
- bmp_image_t, [7](#)

- create_bmp
 - libsurround.h, [11](#)

- err_exit
 - libsurround.h, [12](#)
- err_exit_alt
 - libsurround.h, [12](#)
- err_msg
 - libsurround.h, [13](#)

- filename_from_pathname
 - libsurround.h, [13](#)

- libsurround.h, [9](#)
 - bmp_image_get_height, [10](#)
 - bmp_image_get_pixel, [11](#)
 - bmp_image_get_width, [11](#)
 - create_bmp, [11](#)
 - err_exit, [12](#)
 - err_exit_alt, [12](#)
 - err_msg, [13](#)
 - filename_from_pathname, [13](#)
 - set_pixel, [13](#)
 - usage_err, [15](#)

- set_pixel
 - libsurround.h, [13](#)
- surround_window_t, [7](#)

- usage_err
 - libsurround.h, [15](#)