

Bitmap Writer

Generated by Doxygen 1.8.6

Thu Jul 27 2017 11:33:15

Contents

1	Namespace Index	1
1.1	Namespace List	1
2	Hierarchical Index	3
2.1	Class Hierarchy	3
3	Class Index	5
3.1	Class List	5
4	File Index	7
4.1	File List	7
5	Namespace Documentation	9
5.1	COLOR Namespace Reference	9
5.1.1	Detailed Description	9
6	Class Documentation	11
6.1	Bitmap Class Reference	11
6.1.1	Detailed Description	12
6.1.2	Constructor & Destructor Documentation	12
6.1.2.1	~Bitmap	12
6.1.2.2	Bitmap	12
6.1.2.3	Bitmap	12
6.1.2.4	Bitmap	12
6.1.3	Member Function Documentation	12
6.1.3.1	check	12
6.1.3.2	load	12
6.1.3.3	load	12
6.1.3.4	store	12
6.1.3.5	store	12
6.1.4	Member Data Documentation	13
6.1.4.1	m_data	13
6.1.4.2	m_dib	13

6.1.4.3	m_fname	13
6.1.4.4	m_header	13
6.1.4.5	m_height	13
6.1.4.6	m_rowsize	13
6.1.4.7	m_width	13
6.2	Bitmap::BmpFileHeader_t Struct Reference	13
6.2.1	Detailed Description	13
6.2.2	Member Data Documentation	13
6.2.2.1	bmp_offset	13
6.2.2.2	creator1	13
6.2.2.3	creator2	14
6.2.2.4	filesz	14
6.2.2.5	magic	14
6.3	Bitmap::DibHeader_t Struct Reference	14
6.3.1	Detailed Description	14
6.3.2	Member Data Documentation	14
6.3.2.1	bitspp	14
6.3.2.2	bmp_bytesz	14
6.3.2.3	compress_type	14
6.3.2.4	header_sz	15
6.3.2.5	height	15
6.3.2.6	hres	15
6.3.2.7	ncolors	15
6.3.2.8	nimpcolors	15
6.3.2.9	nplanes	15
6.3.2.10	vres	15
6.3.2.11	width	15
6.4	GreylImage Class Reference	15
6.4.1	Constructor & Destructor Documentation	15
6.4.1.1	GreylImage	15
6.4.1.2	GreylImage	16
6.4.2	Member Function Documentation	16
6.4.2.1	fill	16
6.4.2.2	getPixel	16
6.4.2.3	setPixel	16
6.5	GreyPixel Class Reference	17
6.5.1	Detailed Description	17
6.5.2	Constructor & Destructor Documentation	17
6.5.2.1	GreyPixel	17
6.5.3	Member Data Documentation	17

6.5.3.1	value	17
6.6	RgbImage Class Reference	17
6.6.1	Detailed Description	18
6.6.2	Constructor & Destructor Documentation	18
6.6.2.1	RgbImage	18
6.6.2.2	RgbImage	18
6.6.3	Member Function Documentation	18
6.6.3.1	fill	18
6.6.3.2	getPixel	18
6.6.3.3	setPixel	19
6.7	RgbPixel Class Reference	19
6.7.1	Detailed Description	19
6.7.2	Constructor & Destructor Documentation	19
6.7.2.1	RgbPixel	19
6.7.2.2	RgbPixel	20
6.7.3	Member Data Documentation	20
6.7.3.1	blue	20
6.7.3.2	green	20
6.7.3.3	red	20
7	File Documentation	21
7.1	.dep.inc File Reference	21
7.2	Bitmap.cpp File Reference	21
7.3	Bitmap.h File Reference	21
7.4	main.cpp File Reference	22
7.4.1	Function Documentation	22
7.4.1.1	main	22
	Index	23

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

COLOR	9
-----------------	---

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Bitmap	11
GreylImage	15
RgbImage	17
Bitmap::BmpFileHeader_t	13
Bitmap::DibHeader_t	14
GreyPixel	17
RgbPixel	19

Chapter 3

Class Index

3.1 Class List

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

Bitmap		
Abstract Bitmap class	11
Bitmap::BmpFileHeader_t	13
Bitmap::DibHeader_t	14
GreyImage	15
GreyPixel	17
RgbImage		
Class handling 24-bit RGB Images	17
RgbPixel	19

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

.dep.inc	21
Bitmap.cpp	21
Bitmap.h	21
main.cpp	22

Chapter 5

Namespace Documentation

5.1 COLOR Namespace Reference

5.1.1 Detailed Description

Namespace containing different colour definitions:

- Black
- White
- Grey
- Lightgrey
- Darkgrey
- Blue
- Red
- Green
- Magenta
- Cyan
- Yellow

Chapter 6

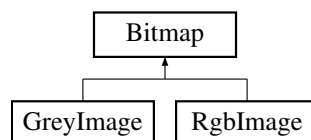
Class Documentation

6.1 Bitmap Class Reference

Abstract [Bitmap](#) class.

```
#include <Bitmap.h>
```

Inheritance diagram for Bitmap:



Classes

- struct [BmpFileHeader_t](#)
- struct [DibHeader_t](#)

Public Member Functions

- virtual [~Bitmap](#) ()
- void [load](#) ()
- void [load](#) (const std::string &fname)
- void [store](#) () const
- void [store](#) (const std::string &fname) const

Protected Member Functions

- [Bitmap](#) ()
- [Bitmap](#) (int32_t width, int32_t height)
- [Bitmap](#) (const std::string &fname)
- void [check](#) (int32_t x, int32_t y) const

Protected Attributes

- [BmpFileHeader_t](#) m_header
- [DibHeader_t](#) m_dib

- `std::string m_fname`
- `int32_t m_width`
- `int32_t m_height`
- `int32_t m_rowsize`
- `uint8_t** m_data`

6.1.1 Detailed Description

Abstract `Bitmap` class.

This class should not be used directly, as it has no sense of different colour depths. Image data is seen on byte level only.

6.1.2 Constructor & Destructor Documentation

6.1.2.1 `Bitmap::~~Bitmap ()` `[virtual]`

Cleans up dynamically generated bitmap array

6.1.2.2 `Bitmap::Bitmap ()` `[protected]`

6.1.2.3 `Bitmap::Bitmap (int32_t width, int32_t height)` `[protected]`

6.1.2.4 `Bitmap::Bitmap (const std::string & fname)` `[protected]`

6.1.3 Member Function Documentation

6.1.3.1 `void Bitmap::check (int32_t x, int32_t y) const` `[protected]`

6.1.3.2 `void Bitmap::load ()`

Loads image data from file, if and only if the member field `m_fname` is set, which happens when using the string constructor. If `m_fname` is empty, this function ends.

6.1.3.3 `void Bitmap::load (const std::string & fname)`

Loads image from file with given name. Throws `invalid_argument` when file could not be opened and `runtime_error` when file is incomplete. No other checks are done.

Parameters

<i>fname</i>	URL to file
--------------	-------------

6.1.3.4 `void Bitmap::store () const`

Stores image data to file, if and only if member field `m_fname` is set, which happens when using the string constructor. If `m_fname` is empty, this function ends.

6.1.3.5 `void Bitmap::store (const std::string & fname) const`

Stores image as bmp to file denoted by `fname`. Existing files will be overwritten

Parameters

<i>fname</i>	URL to file
--------------	-------------

6.1.4 Member Data Documentation

6.1.4.1 `uint8_t** Bitmap::m_data` [protected]

6.1.4.2 `DibHeader_t Bitmap::m_dib` [protected]

6.1.4.3 `std::string Bitmap::m_fname` [protected]

6.1.4.4 `BmpFileHeader_t Bitmap::m_header` [protected]

6.1.4.5 `int32_t Bitmap::m_height` [protected]

6.1.4.6 `int32_t Bitmap::m_rowsize` [protected]

6.1.4.7 `int32_t Bitmap::m_width` [protected]

The documentation for this class was generated from the following files:

- [Bitmap.h](#)
- [Bitmap.cpp](#)

6.2 Bitmap::BmpFileHeader_t Struct Reference

```
#include <Bitmap.h>
```

Public Attributes

- `uint8_t magic` [2]
- `uint32_t filesz`
- `uint16_t creator1`
- `uint16_t creator2`
- `uint32_t bmp_offset`

6.2.1 Detailed Description

`magic`: Header identifier, should be "BM" for Windows NT Bitmaps

`filesz`: complete file size in bytes

`creator1` and `creator2`: reserved

`bmp_offset`: Offset where pixel data array can be found

6.2.2 Member Data Documentation

6.2.2.1 `uint32_t Bitmap::BmpFileHeader_t::bmp_offset`

6.2.2.2 `uint16_t Bitmap::BmpFileHeader_t::creator1`

6.2.2.3 `uint16_t` `Bitmap::BmpFileHeader_t::creator2`

6.2.2.4 `uint32_t` `Bitmap::BmpFileHeader_t::filesize`

6.2.2.5 `uint8_t` `Bitmap::BmpFileHeader_t::magic[2]`

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

- [Bitmap.h](#)

6.3 `Bitmap::DibHeader_t` Struct Reference

```
#include <Bitmap.h>
```

Public Attributes

- `uint32_t` `header_sz`
- `int32_t` `width`
- `int32_t` `height`
- `uint16_t` `nplanes`
- `uint16_t` `bitspp`
- `uint32_t` `compress_type`
- `uint32_t` `bmp_bytesz`
- `int32_t` `hres`
- `int32_t` `vres`
- `uint32_t` `ncolors`
- `uint32_t` `nimpcolors`

6.3.1 Detailed Description

`header_sz`: Size of DIB Header in bytes. Use 40bytes for standard Windows NT DIB Header.

`width`: the bitmap width in pixels (signed integer)

`height`: the bitmap height in pixels (signed integer)

`nplanes`: the number of color planes must be 1

`bitspp`: the number of bits per pixel, which is the color depth of the image. Typical values are 1, 4, 8, 16, 24 and 32.

`compress_type`: Compression method used, 0 for none

`bmp_bytesz`: Size of raw bitmap data, can be 0 if no compression is used

`hres` and `vres`: horizontal and vertical resolution in pixel per meter

`ncolors`: the number of colours in the colour palette

`nimpcolor`: the number of important colours, 0 for every colour is important

6.3.2 Member Data Documentation

6.3.2.1 `uint16_t` `Bitmap::DibHeader_t::bitspp`

6.3.2.2 `uint32_t` `Bitmap::DibHeader_t::bmp_bytesz`

6.3.2.3 `uint32_t` `Bitmap::DibHeader_t::compress_type`

6.3.2.4 `uint32_t Bitmap::DibHeader_t::header_sz`

6.3.2.5 `int32_t Bitmap::DibHeader_t::height`

6.3.2.6 `int32_t Bitmap::DibHeader_t::hres`

6.3.2.7 `uint32_t Bitmap::DibHeader_t::ncolors`

6.3.2.8 `uint32_t Bitmap::DibHeader_t::nimpcolors`

6.3.2.9 `uint16_t Bitmap::DibHeader_t::nplanes`

6.3.2.10 `int32_t Bitmap::DibHeader_t::vres`

6.3.2.11 `int32_t Bitmap::DibHeader_t::width`

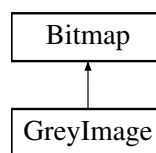
The documentation for this struct was generated from the following file:

- [Bitmap.h](#)

6.4 GreylImage Class Reference

```
#include <Bitmap.h>
```

Inheritance diagram for GreylImage:



Public Member Functions

- [GreylImage](#) (`int32_t width`, `int32_t height`)
- [GreylImage](#) (`const std::string &fname`)
- [GreyPixel](#) `getPixel` (`int32_t x`, `int32_t y`) `const`
- `void` [setPixel](#) (`int32_t x`, `int32_t y`, [GreyPixel](#) `val`)
- `void` [fill](#) ([GreyPixel](#) `p`)

Additional Inherited Members

6.4.1 Constructor & Destructor Documentation

6.4.1.1 `GreylImage::GreylImage (int32_t width, int32_t height)`

Creates an instance with given width and height in pixels. The proper amount of memory is allocated, but not filled, so image data is kind of random at this point. This throws an `invalid_argument` exception if width and/or height is negative. Note that the standard actually supports negative height, but this implementation does not.

Note that also this uses a LUT, there is currently no way to edit the LUT. The LUT is created when the image is stored as a .bmp file and contains 256 entries for the 256 Greyscale colours.

Parameters

<i>width</i>	Desired image width in pixels
<i>height</i>	Desired image height in pixels

6.4.1.2 `GreylImage::GreylImage (const std::string & fname)`

Creates an instance by loading from the file given as parameter. This uses the [Bitmap](#) class' method [load\(\)](#) to load that file.

If the file contained a LUT, this LUT is not loaded to memory.

Parameters

<i>fname</i>	URL to file
--------------	-------------

6.4.2 Member Function Documentation

6.4.2.1 `void GreylImage::fill (GreyPixel p)`

This fills the whole image by calling `setPixel` for every valid (x,y) combination.

Parameters

<i>p</i>	GreyPixel element that should be written
----------	--

6.4.2.2 `GreyPixel GreylImage::getPixel (int32_t x, int32_t y) const`

Gets a pixel from the [Bitmap](#). Throws `out_of_range` if the desired pixel is not inside the actual [Bitmap](#).

Parameters

<i>x</i>	X position, 0 is left
<i>y</i>	Y position, 0 is up

Returns

Copy of [GreyPixel](#) Element at (x,y)

6.4.2.3 `void GreylImage::setPixel (int32_t x, int32_t y, GreyPixel val)`

Sets the desired pixel to the given value. Throws `out_of_range` and does not change anything if the desired location is outside the actual image.

Parameters

<i>x</i>	X position, 0 is left
<i>y</i>	Y position, 0 is up
<i>val</i>	GreyPixel element that should be written

The documentation for this class was generated from the following files:

- [Bitmap.h](#)
- [Bitmap.cpp](#)

6.5 GreyPixel Class Reference

```
#include <Bitmap.h>
```

Public Member Functions

- [GreyPixel](#) (uint8_t *value*)

Public Attributes

- uint8_t *value*

6.5.1 Detailed Description

Class holding single byte value denoting grey values ranging from 0 to 255.

6.5.2 Constructor & Destructor Documentation

6.5.2.1 GreyPixel::GreyPixel (uint8_t *value*) [inline]

Constructs [GreyPixel](#) and sets internal value to given value

Parameters

<i>value</i>	Grey value ranging from 0 to 255, 0 beeing black
--------------	--

6.5.3 Member Data Documentation

6.5.3.1 uint8_t GreyPixel::value

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

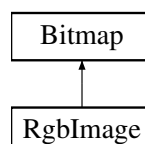
- [Bitmap.h](#)

6.6 RgbImage Class Reference

Class handling 24-bit RGB Images.

```
#include <Bitmap.h>
```

Inheritance diagram for RgbImage:



Public Member Functions

- [RgbImage](#) (int32_t width, int32_t height)
- [RgbImage](#) (const std::string &fname)

- [RgbPixel](#) `getPixel` (int32_t x, int32_t y) const
- void [setPixel](#) (int32_t x, int32_t y, [RgbPixel](#) val)
- void [fill](#) ([RgbPixel](#) p)

Additional Inherited Members

6.6.1 Detailed Description

Class handling 24-bit RGB Images.

This class handles a single [Bitmap](#) image with a colour depth of 24 bit, meaning three byte values for red, green and blue channel. An Alpha Channel is not supported.

6.6.2 Constructor & Destructor Documentation

6.6.2.1 `RgbImage::RgbImage (int32_t width, int32_t height)`

Creates an instance with given width and height in pixels. The proper amount of memory is allocated, but not filled, so image data is kind of random at this point. This throws an `invalid_argument` exception if width and/or height is negative. Note that the standard actually supports negative height, but this implementation does not.

Parameters

<i>width</i>	Desired image width in pixels
<i>height</i>	Desired image height in pixels

6.6.2.2 `RgbImage::RgbImage (const std::string & fname)`

Creates an instance by loading from the file given as parameter. This uses the [Bitmap](#) class' method `load()` to load that file.

Parameters

<i>fname</i>	URL to file
--------------	-------------

6.6.3 Member Function Documentation

6.6.3.1 `void RgbImage::fill (RgbPixel p)`

This fills the whole image by calling `setPixel` for every valid (x,y) combination.

Parameters

<i>p</i>	RgbPixel element that should be written
----------	---

6.6.3.2 `RgbPixel RgbImage::getPixel (int32_t x, int32_t y) const`

Gets a pixel from the [Bitmap](#). Throws `out_of_range` if the desired pixel is not inside the actual [Bitmap](#).

Parameters

<i>x</i>	X position, 0 is left
----------	-----------------------

<i>y</i>	Y position, 0 is up
----------	---------------------

Returns

Copy of [RgbPixel](#) Element at (x,y)

6.6.3.3 void RgbImage::setPixel (int32_t x, int32_t y, RgbPixel val)

Sets the desired pixel to the given value. Throws out_of_range and does not change anything if the desired location is outside the actual image.

Parameters

<i>x</i>	X position, 0 is left
<i>y</i>	Y position, 0 is up
<i>val</i>	RgbPixel element that should be written

The documentation for this class was generated from the following files:

- [Bitmap.h](#)
- [Bitmap.cpp](#)

6.7 RgbPixel Class Reference

```
#include <Bitmap.h>
```

Public Member Functions

- [RgbPixel](#) (const [GreyPixel](#) &p)
- [RgbPixel](#) (uint8_t [blue](#), uint8_t [green](#), uint8_t [red](#))

Public Attributes

- uint8_t [red](#)
- uint8_t [green](#)
- uint8_t [blue](#)

6.7.1 Detailed Description

Class holding three byte values denoting red, green and blue values for creating a RGB-Value.

6.7.2 Constructor & Destructor Documentation

6.7.2.1 RgbPixel::RgbPixel (const GreyPixel & p) [inline]

Conversion from [GreyPixel](#). Sets alls red, green and blue values to grey value from given [GreyPixel](#)

Parameters

<i>p</i>	GreyPixel to be copied
----------	--

6.7.2.2 `RgbPixel::RgbPixel (uint8_t blue, uint8_t green, uint8_t red)` `[inline]`

Instantiates a [RgbPixel](#) with given red, green and blue values

Parameters

<i>blue</i>	Value of blue chanel from 0 to 255, 0 meaning black
<i>green</i>	Value of green chanel from 0 to 255, 0 meaning black
<i>red</i>	Value of green chanel from 0 to 255, 0 meaning black

6.7.3 Member Data Documentation

6.7.3.1 `uint8_t RgbPixel::blue`

6.7.3.2 `uint8_t RgbPixel::green`

6.7.3.3 `uint8_t RgbPixel::red`

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

- [Bitmap.h](#)

Chapter 7

File Documentation

7.1 .dep.inc File Reference

7.2 Bitmap.cpp File Reference

```
#include "Bitmap.h"  
#include <fstream>  
#include <stdexcept>
```

7.3 Bitmap.h File Reference

```
#include <stdint.h>  
#include <iostream>  
#include <string>
```

Classes

- class [Bitmap](#)
Abstract [Bitmap](#) class.
- struct [Bitmap::BmpFileHeader_t](#)
- struct [Bitmap::DibHeader_t](#)
- class [GreyPixel](#)
- class [RgbPixel](#)
- class [RgbImage](#)
Class handling 24-bit RGB Images.
- class [GreyImage](#)

Namespaces

- [COLOR](#)

7.4 main.cpp File Reference

```
#include <cstdlib>
#include <iostream>
#include "Bitmap.h"
#include <fstream>
```

Functions

- int [main](#) (int argc, char **argv)

7.4.1 Function Documentation

7.4.1.1 int main (int *argc*, char ** *argv*)

Index

- ~Bitmap
 - Bitmap, [12](#)
- .dep.inc, [21](#)
- Bitmap, [11](#)
 - ~Bitmap, [12](#)
 - Bitmap, [12](#)
 - check, [12](#)
 - load, [12](#)
 - m_data, [13](#)
 - m_dib, [13](#)
 - m_fname, [13](#)
 - m_header, [13](#)
 - m_height, [13](#)
 - m_rowsize, [13](#)
 - m_width, [13](#)
 - store, [12](#)
- Bitmap.cpp, [21](#)
- Bitmap.h, [21](#)
- Bitmap::BmpFileHeader_t, [13](#)
 - bmp_offset, [13](#)
 - creator1, [13](#)
 - creator2, [13](#)
 - filesz, [14](#)
 - magic, [14](#)
- Bitmap::DibHeader_t, [14](#)
 - bitspp, [14](#)
 - bmp_bytesz, [14](#)
 - compress_type, [14](#)
 - header_sz, [14](#)
 - height, [15](#)
 - hres, [15](#)
 - ncolors, [15](#)
 - nimpcolors, [15](#)
 - nplanes, [15](#)
 - vres, [15](#)
 - width, [15](#)
- bitspp
 - Bitmap::DibHeader_t, [14](#)
- blue
 - RgbPixel, [20](#)
- bmp_bytesz
 - Bitmap::DibHeader_t, [14](#)
- bmp_offset
 - Bitmap::BmpFileHeader_t, [13](#)
- COLOR, [9](#)
- check
 - Bitmap, [12](#)
- compress_type
 - Bitmap::DibHeader_t, [14](#)
- creator1
 - Bitmap::BmpFileHeader_t, [13](#)
- creator2
 - Bitmap::BmpFileHeader_t, [13](#)
- filesz
 - Bitmap::BmpFileHeader_t, [14](#)
- fill
 - GreyImage, [16](#)
 - RgbImage, [18](#)
- getPixel
 - GreyImage, [16](#)
 - RgbImage, [18](#)
- green
 - RgbPixel, [20](#)
- GreyImage, [15](#)
 - fill, [16](#)
 - getPixel, [16](#)
 - GreyImage, [15](#), [16](#)
 - GreyImage, [15](#), [16](#)
 - setPixel, [16](#)
- GreyPixel, [17](#)
 - GreyPixel, [17](#)
 - GreyPixel, [17](#)
 - value, [17](#)
- header_sz
 - Bitmap::DibHeader_t, [14](#)
- height
 - Bitmap::DibHeader_t, [15](#)
- hres
 - Bitmap::DibHeader_t, [15](#)
- load
 - Bitmap, [12](#)
- m_data
 - Bitmap, [13](#)
- m_dib
 - Bitmap, [13](#)
- m_fname
 - Bitmap, [13](#)
- m_header
 - Bitmap, [13](#)
- m_height
 - Bitmap, [13](#)
- m_rowsize
 - Bitmap, [13](#)
- m_width
 - Bitmap, [13](#)

- Bitmap, [13](#)
- magic
 - Bitmap::BmpFileHeader_t, [14](#)
- main
 - main.cpp, [22](#)
- main.cpp, [22](#)
 - main, [22](#)
- ncolors
 - Bitmap::DibHeader_t, [15](#)
- nimpcolors
 - Bitmap::DibHeader_t, [15](#)
- nplanes
 - Bitmap::DibHeader_t, [15](#)
- red
 - RgbPixel, [20](#)
- RgbImage, [17](#)
 - fill, [18](#)
 - getPixel, [18](#)
 - RgbImage, [18](#)
 - RgbImage, [18](#)
 - setPixel, [19](#)
- RgbPixel, [19](#)
 - blue, [20](#)
 - green, [20](#)
 - red, [20](#)
 - RgbPixel, [19](#), [20](#)
 - RgbPixel, [19](#), [20](#)
- setPixel
 - GreyImage, [16](#)
 - RgbImage, [19](#)
- store
 - Bitmap, [12](#)
- value
 - GreyPixel, [17](#)
- vres
 - Bitmap::DibHeader_t, [15](#)
- width
 - Bitmap::DibHeader_t, [15](#)