libCbg

Develop Docs

Documents Author(English version): copper187

Date: 2020-10-30

Version: Ver.20y44w05d(build 2)

© copper187.

All rights reserved.

This software is licensed under GNU LGPL v3 license.

Use and distribution this software, You must compliance the copyleft restrictions that the LGPL v3 license imposes.

Contents

Functions:

cbg1_enc_default()----------4

cbg1_enc_advanced()----------5

cbg1_encToFile_default()----------7

cbg1_encToFile_advanced()----------8

cbg1_get_trans_pixel()----------10

cbg1_get_huffman_stream()----------11

Struct:

cbg_codec::api*----------12

```
cbg_codec::api* cbg1_enc_default
( int height,
  int width,
  int color_depth,
  unsigned char/BYTE *raw_pixel_buffer)
```

Explanation:

- Encode to cbg image format by pixels in raw pixel buffer and use height, width and depth you pass. Write the cbg bit stream (include “CompressedBG___” magic bytes and all file header information.) in a bitstream buffer.
- libcbg will return a “cbg_codec::api*” type struct pointer. Change to P15 to read more about this struct.
- libcbg will use default key (0x31676263 or “cbg1”) and write default encoder information (“bylibcbg”).
- libcbg will use default huffman coding settings(multithreads).

Error codes:

- 0x1: Height or width are wrong. (negative or zero).

  Tips: check height and width and pass a correct num.
- 0x2: Wrong or not support color depth.

  Tips: check color depth and pass a correct num.

```
cbg_codec::api* cbg1_enc_advanced
( int height,
  int width,
  int color_depth,
  unsigned int/DWORD key,
  char *encoder_information,
  bool huffman_coding_settings,
  unsigned char/BYTE *raw_pixel_buffer )
```

Explanation:

- Encode to cbg image format by pixels in raw pixel buffer and use height, width, depth and key you pass. Write the cbg bit stream(include "CompressedBG___" magic bytes and all file header information.) in a bitstream buffer.
- libcbg will return a "cbg_codec::api*" type struct pointer. Change to P15 to read more about this struct.
- libcbg will write encoder information you pass.
- libcbg will use singlethread if the huffman coding settings is false, and will use multithreads if it is true.

  Tips: libcbg will not use less memory if you use singlethread coding. Maybe support in next version? (maybe).

Error codes:

- 0x1: Height or width are wrong.(negative or zero).

Tips: check height and width and pass a correct num.

- 0x1: Wrong or not support color depth.

  Tips: check color depth and pass a correct num.

Warning codes:

- 0x101: Encoder information is too long(Out of 8 bytes).

  Tips: libcbg will only be use first 8 bytes character.

```
int cbg1_encToFile_default
( int height,
  int width,
  int color_depth,
  unsigned char/BYTE *raw_pixel_buffer,
  char *filename)
```

Explanation:

- Encode to cbg image format by pixels in raw pixel buffer and use height, width and depth you pass. Write the cbg bit stream (include “CompressedBG___” magic bytes and all file header information.) in bit stream buffer. Write the bit stream in a file you designated and only return a successed code(0x0).
- libcbg will use default key (0x31676263 or “cbg1”) and write default encoder information (“bylibcbg”).
- libcbg will use default huffman coding settings(multithreads).

Error codes:

- 0x1: Height or width are wrong.(negative or zero).

  Tips: check height and width and pass a correct num.
- 0x2: Wrong or not support color depth.

  Tips: check color depth and pass a correct num.

```
int cbg1_encToFile_advanced
( int height,
  int width,
  int color_depth,
  unsigned int/DWORD key,
  char *encoder_information,
  bool huffman_coding_settings,
  unsigned char/BYTE *raw_pixel_buffer,
  char *filename)
```

Explanation:

- Encode to cbg image format by pixels in raw pixel buffer and use height, width, depth and key you pass. Write the cbg bit stream(include “CompressedBG___” magic bytes and all file header information.) in bit stream buffer. Write the bit stream in a file you designated and only return a successed code(0x0).
- libcbg will write encoder information you pass.
- libcbg will use singlethread if the huffman coding settings is false, and will use multithreads if it is true.

  Tips: libcbg will not use less memory if you use singlethread coding. Maybe support in next version? (maybe).

Error codes:

- 0x1: Height or width are wrong.(negative or zero).

Tips: check height and width and pass a correct num.

- 0x2: Wrong or not support color depth.

  Tips: check color depth and pass a correct num.

Warning codes:

- 0x101: Encoder information is too long(Out of 8 bytes).

  Tips: libcbg will only be use first 8 bytes character.

```
cbg_codec::api* cbg1_get_trans_pixel
( int height,
  int width,
  int color_depth,
  unsigned char/BYTE *raw_pixel_buffer)
```

Explanation:

- Transform the pixels from raw pixel buffer to cbg format used pixels. Use height, width and depth you pass. Write the transformed pixel stream in a bitstream buffer.
- libcbg will return a “cbg_codec::api*” type struct pointer. Change to P15 to read more about this struct.

Error codes:

- 0x1: Height or width are wrong.(negative or zero).

  Tips: check height and width and pass a correct num.
- 0x2: Wrong or not support color depth.

  Tips: check color depth and pass a correct num.

```
cbg_codec::api* cbg1_get_huffman_stream
( int height,
  int width,
  int color_depth,
  unsigned char/BYTE *raw_pixel_buffer)
```

Explanation:

- Encode the pixels to cbg format style huffman coding(cbg format is using a special huffman tree).Use height, width and depth you pass. Write the huffman coding bit stream in a bitstream buffer.
- libcbg will return a “cbg_codec::api*” type struct pointer. Change to P15 to read more about this struct.
- libcbg will use default huffman coding settings(multithreads).

Error codes:

- 0x1: Height or width are wrong. (negative or zero).

  Tips: check height and width and pass a correct num.
- 0x2: Wrong or not support color depth.

  Tips: check color depth and pass a correct num.

```
cbg_codec::api*
{
    unsigned long long buffersize;
    unsigned char buffer[];
}
```

- `unsigned long long buffersize;`

  The size of this buffer.

- `unsigned char buffer[];`

  The binary stream buffer pointer. Size of this buffer is recorded in “unsigned long long buffersize”.