

---

# **libaescprintf Documentation**

***Release 1.0.0***

**Brüggemann Eddie**

**Sep 08, 2017**



## CONTENTS:

<b>1</b>	<b>libaescprintf (ANSI Escape Sequence Color Print Format).</b>	<b>1</b>
1.1	Introduction . . . . .	1
1.2	libaescprintf . . . . .	1
<b>2</b>	<b>Types of libaescprintf</b>	<b>3</b>
2.1	Styles . . . . .	4
2.2	Colors . . . . .	4
<b>3</b>	<b>Defines of libaescprintf</b>	<b>7</b>
<b>4</b>	<b>The styled colored printf () functions:</b>	<b>9</b>
4.1	Styled Print Format: . . . . .	9
4.2	Color Foreground Print Format: . . . . .	9
4.3	Color Background Print Format: . . . . .	9
4.4	Styled Foreground Color Print Format: . . . . .	10
4.5	Styled Background Color Print Format: . . . . .	10
4.6	Color Foreground Background Print Format: . . . . .	10
4.7	Styled Color Foreground Background Print Format: . . . . .	10
<b>5</b>	<b>The styled colored fprintf () functions</b>	<b>13</b>
5.1	File Styled Print Format: . . . . .	13
5.2	File Color Foreground Print Format: . . . . .	13
5.3	File Color Background Print Format: . . . . .	13
5.4	File Styled Foreground Color Print Format: . . . . .	14
5.5	Styled Background Color Print Format: . . . . .	14
5.6	File Color Foreground Background Print Format: . . . . .	14
5.7	File Styled Color Foreground Background Print Format: . . . . .	15
<b>6</b>	<b>About libaescprintf</b>	<b>17</b>
6.1	Compatibility . . . . .	17
6.2	Note from the author . . . . .	17
<b>7</b>	<b>libaescprintf screen-shots:</b>	<b>19</b>
7.1	Matrix output . . . . .	19
7.2	Text output . . . . .	20
7.3	Flags output . . . . .	21
<b>8</b>	<b>Indices and tables</b>	<b>23</b>
	<b>Index</b>	<b>25</b>



## LIBAESCPRINTF (ANSI ESCAPE SEQUENCE COLOR PRINT FORMAT).

**library** libaescprintf

**version** 1.0.0

**author** Brüggemann Eddie

**contact** <[mrcyberfighter@gmail.com](mailto:mrcyberfighter@gmail.com)>

**license** LGPLv3

**website** <<http://www.open-source-projects.net/libaescprintf/libaescprintf>>

**release** Sep 08, 2017

### 1.1 Introduction

**libaescprintf** is a little library based on the ANSI escape sequences and only compatible with terminals *ANSI escape sequences* compliant terminals.

Most of terminal that I have tested works well but not all.

But in addition **libaescprintf** provides

some styles features like **bold**, ~~**strike-trough**~~, *italic*, and so on...

---

**Note:** You can insert the *ANSI escapes sequences*

into an **echo** command or using the binary **tput**

but with this library you can use it into a source code.

---

**note** *The goal* of this library is to provide a lightweight textual pretty output.

### 1.2 libaescprintf

- **libaescprintf** can be divided into 2 different output functions types.

- `printf(...)` ()

For simply formatting the output.

and

- `fprintf(...)` ()

For using the `FILE` \* **stream** you want.

And why not serializing to a file...

- **libaescprintf** can be divided into *background and foreground* colors.

The colors can be divided into Light and Dark colors.

- **libaescprintf** provides styles features.

## TYPES OF LIBAESCPRINTF

**libaescprintf** make usage of different types of colors and styles:

```
// Building a color and a style type. (Used into library functions).

enum style_enum { normal = 0 , bold = 1 , dim = 2 , italic = 3, underline = 4 , blink = 5, inverted = 7, hidden = 8, strikethrough = 9 } ;

//typedef enum color_enum color_t ;
typedef enum style_enum style_t ;

// Building a foreground color type. (Used into library functions).

// Foreground Normal >= 30 <= 37 (8 colors)
// Foreground Light >= 90 <= 97 (8 colors)
enum fg_color_enum { fg_black = 30, fg_darkgray = 90,
fg_red = 31, fg_lightrd = 91,
fg_green = 32, fg_lightgreen = 92,
fg_yellow = 33, fg_lightyellow = 93,
fg_blue = 34, fg_lightblue = 94,
fg_pink = 35, fg_lightpink = 95,
fg_cyan = 36, fg_lightcyan = 96,
fg_lightgray = 37, fg_white = 97,

fg_default = 39 } ;

// Building a background color type. (Used into library functions).

// Background Normal >= 40 <= 47 (8 colors)
// Background Light >= 100 <= 107 (8 colors)
enum bg_color_enum { bg_black = (30+10), bg_darkgray = (90+10),
bg_red = (31+10), bg_lightrd = (91+10),
bg_green = (32+10), bg_lightgreen = (92+10),
bg_yellow = (33+10), bg_lightyellow = (93+10),
bg_blue = (34+10), bg_lightblue = (94+10),
bg_pink = (35+10), bg_lightpink = (95+10),
bg_cyan = (36+10), bg_lightcyan = (96+10),
bg_lightgray = (37+10), bg_white = (97+10),

bg_default = (39+10) } ;

typedef enum fg_color_enum fg_color_t ;
typedef enum bg_color_enum bg_color_t ;
```

Which you can use in various functions from this library.

## 2.1 Styles

- normal
- bold
- dim
- italic
- underline
- blink
- inverted
- hidden
- strikethrough

## 2.2 Colors

### 2.2.1 Foreground colors values

- fg\_black
- fg\_darkgray
- fg\_red
- fg\_lightred
- fg\_green
- fg\_lightgreen
- fg\_yellow
- fg\_lightyellow
- fg\_blue
- fg\_lightblue
- fg\_pink
- fg\_lightpink
- fg\_cyan
- fg\_lightcyan
- fg\_lightgray
- fg\_white
- fg\_default (Your current terminal foreground color).



### 2.2.2 Background colors values

- `bg_black`
- `bg_darkgray`
- `bg_red`
- `bg_lightred`
- `bg_green`
- `bg_lightgreen`
- `bg_yellow`
- `bg_lightyellow`
- `bg_blue`
- `bg_lightblue`
- `bg_pink`
- `bg_lightpink`
- `bg_cyan`
- `bg_lightcyan`
- `bg_lightgray`
- `bg_white`
- `bg_default` (Your current terminal background color).



## DEFINES OF LIBAESCPRINTF

The ANSI escape sequences are defines as well into **libaescprintf**:

```
/** You can use this defines into raw string formatting. */  
  
#define ESC    "\033" // Escape sequence (octal format).  
  
// Dark colors defines.  
#define DB     ESC "[30m" // Dark Black.  
#define DR     ESC "[31m" // Dark Red.  
#define DG     ESC "[32m" // Dark Green.  
#define DY     ESC "[33m" // Dark Yellow.  
#define DS     ESC "[34m" // Dark Sky. (Dark blue)  
#define DP     ESC "[35m" // Dark Pink.  
#define DC     ESC "[36m" // Dark Cyan.  
#define DW     ESC "[37m" // Dark white. (Light Gray)  
  
// Light colors defines.  
#define LB     ESC "[90m" // Light Black.  
#define LR     ESC "[91m" // Light Red.  
#define LG     ESC "[92m" // Light Green.  
#define LY     ESC "[93m" // Light Yellow.  
#define LS     ESC "[94m" // Light Sky. (Light blue)  
#define LP     ESC "[95m" // Light Pink.  
#define LC     ESC "[96m" // Light Cyan.  
#define LW     ESC "[97m" // Light White.  
  
#define DD     ESC "[39m" // Default color.  
  
// Styles defines  
#define RESET      ESC "[0m" // Reset.  
#define BOLD       ESC "[1m" // Bold.  
#define DIM        ESC "[2m" // DIM  
#define ITALIC     ESC "[3m" // Italic.  
#define UNDERSTRIKE ESC "[4m" // Under-strike.  
#define BLINK      ESC "[5m" // Blink (Doesn't work on all terminals.  
↳look at xterm it work's on it !).  
#define INVERTED   ESC "[7m" // Inverted (Invert foreground and  
↳background).  
#define HIDDEN     ESC "[8m" // Hidden (Print only background color).  
#define STRIKETROUGH ESC "[9m" // Strike-through.
```

You can simply make usage of them into `printf()` family functions.



## THE STYLED COLORED PRINTF () FUNCTIONS:

### 4.1 Styled Print Format:

int **s\_printf** (style\_t *style*, const char \**fmt*, ...)

**Parameters**

- **style** – the *style* to apply to the text.
- **fmt** – The format string for the output.

**Returns** The number of characters from the output string or -1 in case of error.

**Return type** int.

### 4.2 Color Foreground Print Format:

int **cfg\_printf** (fg\_color\_t *color*, const char \**fmt*, ...)

**Parameters**

- **fg\_color** – The *foreground color* to apply.
- **fmt** – The format string for the output.

**Returns** The number of characters from the output string or -1 in case of error.

**Return type** int.

### 4.3 Color Background Print Format:

int **cbg\_printf** (bg\_color\_t *bgcolor*, const char \**fmt*, ...)

**Parameters**

- **bg\_color** – The *foreground color* to apply.
- **fmt** – The format string for the output.

**Returns** The number of characters from the output string or -1 in case of error.

**Return type** int.

## 4.4 Styled Foreground Color Print Format:

int **scfg\_printf** (style\_t *style*, fg\_color\_t *fg\_color*, const char *\*fmt*, ...)

### Parameters

- **style** – the *style* to apply to the text.
- **fg\_color** – The *foreground color* to apply.
- **fmt** – The format string for the output.

**Returns** The number of characters from the output string or -1 in case of error.

**Return type** int.

## 4.5 Styled Background Color Print Format:

int **scbg\_printf** (style\_t *style*, bg\_color\_t *bg\_color*, const char *\*fmt*, ...)

### Parameters

- **style** – the *style* to apply to the text.
- **bg\_color** – The *background color* to apply.
- **fmt** – The format string for the output.

**Returns** The number of characters from the output string or -1 in case of error.

**Return type** int.

## 4.6 Color Foreground Background Print Format:

int **cfgbg\_printf** (fg\_color\_t *fg\_color*, bg\_color\_t *bg\_color*, const char *\*fmt*, ...)

### Parameters

- **fg\_color** – The *foreground color* to apply.
- **bg\_color** – The *foreground color* to apply.
- **fmt** – The format string for the output.

**Returns** The number of characters from the output string or -1 in case of error.

**Return type** int.

## 4.7 Styled Color Foreground Background Print Format:

int **scfgbg\_printf** (style\_t *style*, fg\_color\_t *fg\_color*, bg\_color\_t *bg\_color*, const char *\*fmt*, ...)

### Parameters

- **style** – the *style* to apply to the text.
- **fg\_color** – The *foreground color* to apply.
- **bg\_color** – The *foreground color* to apply.

- **fmt** – The format string for the output.

**Returns** The number of characters from the output string or -1 in case of error.

**Return type** `int`.





## THE STYLED COLORED `FPRINTF()` FUNCTIONS

### 5.1 File Styled Print Format:

int **s\_fprintf** (FILE \**fp*, style\_t *style*, const char \**fmt*, ...)

**Parameters**

- **fp** – A FILE pointer.
- **style** – the *style* to apply to the text.
- **fmt** – The format string for the output.

**Returns** The number of characters from the output string or -1 in case of error.

**Return type** int.

### 5.2 File Color Foreground Print Format:

int **cfg\_fprintf** (FILE \**fp*, fg\_color\_t *color*, const char \**fmt*, ...)

**Parameters**

- **fp** – A FILE pointer.
- **fg\_color** – The *foreground color* to apply.
- **fmt** – The format string for the output.

**Returns** The number of characters from the output string or -1 in case of error.

**Return type** int.

### 5.3 File Color Background Print Format:

int **cbg\_fprintf** (FILE \**fp*, bg\_color\_t *bcolor*, const char \**fmt*, ...)

**Parameters**

- **fp** – A FILE pointer.
- **bg\_color** – The *background color* to apply.
- **fmt** – The format string for the output.

**Returns** The number of characters from the output string or -1 in case of error.

**Return type** `int`.

## 5.4 File Styled Foreground Color Print Format:

`int scfg_fprintf (FILE *fp, style_t style, fg_color_t fg_color, const char *fmt, ...)`

### Parameters

- **fp** – A `FILE` pointer.
- **style** – the *style* to apply to the text.
- **fg\_color** – The *foreground color* to apply.
- **fmt** – The format string for the output.

**Returns** The number of characters from the output string or -1 in case of error.

**Return type** `int`.

## 5.5 Styled Background Color Print Format:

`int scbg_printf (FILE *fp, style_t style, bg_color_t bg_color, const char *fmt, ...)`

### Parameters

- **fp** – A `FILE` pointer.
- **style** – the *style* to apply to the text.
- **bg\_color** – The *background color* to apply.
- **fmt** – The format string for the output.

**Returns** The number of characters from the output string or -1 in case of error.

**Return type** `int`.

## 5.6 File Color Foreground Background Print Format:

`int cfgbg_fprintf (FILE *fp, fg_color_t fg_color, bg_color_t bg_color, const char *fmt, ...)`

### Parameters

- **fp** – A `FILE` pointer.
- **fg\_color** – The *foreground color* to apply.
- **bg\_color** – The *background color* to apply.
- **fmt** – The format string for the output.

**Returns** The number of characters from the output string or -1 in case of error.

**Return type** `int`.

## 5.7 File Styled Color Foreground Background Print Format:

int **scfgbg\_fprintf** (FILE *\*fp*, style\_t *style*, fg\_color\_t *fg\_color*, bg\_color\_t *bg\_color*, const char *\*fmt*, ...)

### Parameters

- **fp** – A FILE pointer.
- **style** – the *style* to apply to the text.
- **fg\_color** – The *foreground color* to apply.
- **bg\_color** – The *foreground color* to apply.
- **fmt** – The format string for the output.

**Returns** The number of characters from the output string or -1 in case of error.

**Return type** int.



## ABOUT LIBAESCPRINTF

At first I often try myself to find the generic function into termcap to get the colors and styles codes for building a library compliant with any terminal and I fail to find it...

So I often try to get some help from a terminal dinosaur which master termcap or termios to make a more generic library.

But nobody **HOME**...

---

**Note:** I can do more with terminal Sequences as displaying a True Color image (RGB) into a terminal (xterm, terminals build with libvte-2.91 on GNU/Linux)

But I must reduce the cursor to the minimum size for doing this but I can't restore a normal cursor size...

**enjoy** So enjoy using **libaescprintf** for ANSI art as I enjoy to write this little library.

---

### 6.1 Compatibility

**libaescprintf** was tested on as many terminals I could find into the repositories, sometimes some terminals either don't blink or don't accept **ANSI** escape sequences at all.

But most of the time it works well.

It works in most of the terminals: konsole as gnome-terminal and many others.

But the old bug concern the background and the linefeed appears on some terminals:

if you set the background color other than default and set an linefeed the entire line is background highlighted with the chosen color per example the Apple-Terminal do this one (but the blinking effect works well with it).

### 6.2 Note from the author

I don't try to reinvent the wheel but simply provide this minimalist library in the following goals:

1. At first to provide a lightweight library for only textual effect.

2. To catch your attention so that someone can help me to make the library universal, using termcap (or termios).

## LIBAESCPRINTF SCREEN-SHOTS:

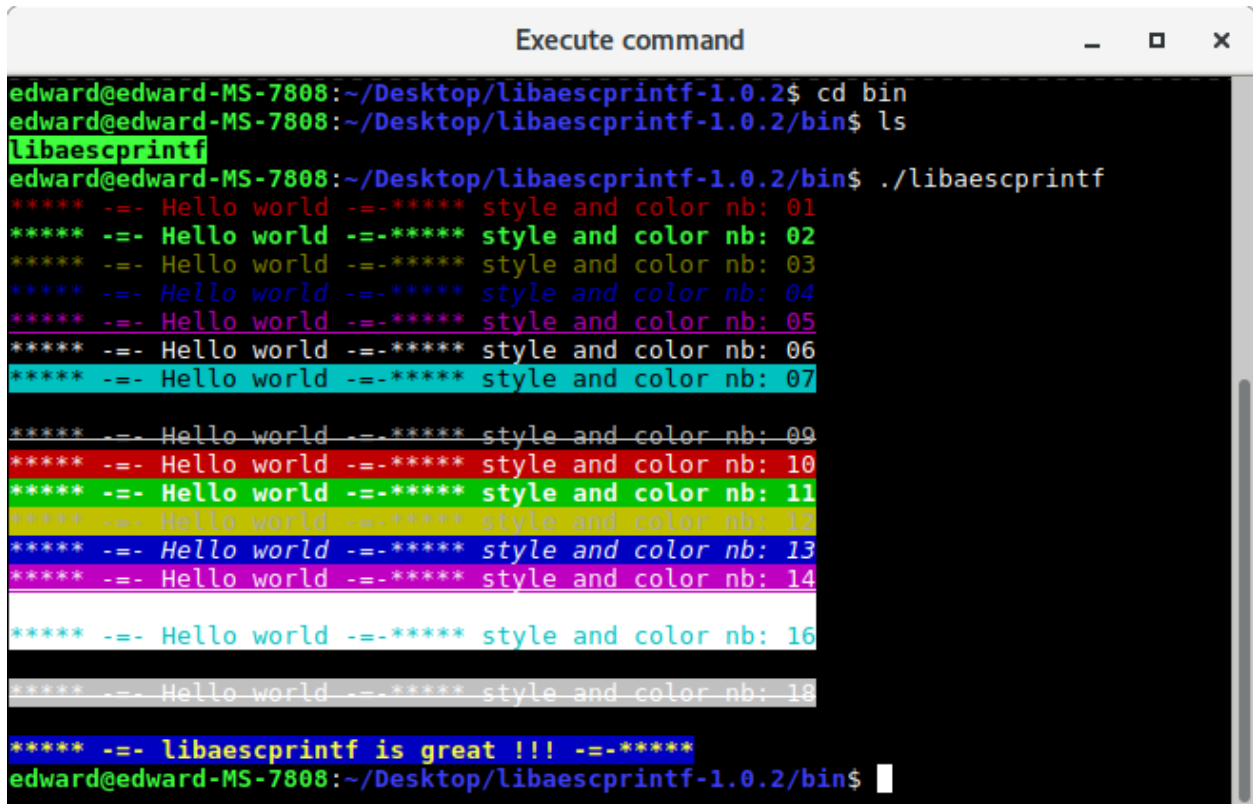
### 7.1 Matrix output



The screenshot shows a terminal window with the title "Execute command". The terminal displays a large matrix of numbers, with each row containing 20 numbers. The numbers are color-coded using a rainbow gradient, where different colors represent different ranges of values. The matrix is displayed on a black background. The numbers are arranged in a grid, with each row starting on a new line. The colors transition from red for the lowest values to purple for the highest values.

```
76960483186840589000966279886216790270355848952302535911248949626197254276022669
56033869893876671015376249754098551555123408410876767017586399695600683526328719
44244710958922751553516311991771331086811175499103290680335252401917746873460465
27784282424804186831386101516474194261251905895091235502113280905560537905913464
83920240315565560246151152936989829097686494503826335624667470443009683738757881
68737132795067384986370725518805559031979228533274305630590955700214755410216180
98816733909341759094084299057933197766886713450165769737481676421715792255367947
08930912472094337589442677839337935827248507130969834772479575290200241171478320
32392142203133098183887999178245709996489036812158671412700177221727910387414590
22305653753508129088198348099185335589836498207216989723662880402993675121639286
86455146330627048935023518832486231787082723409118456472331240169917089990188136
67378482807709973593094976223890211025237726533828400737947288196981668212839908
81170909125981989549073607747977087303820609879498363910649647289909955729611846
77528964141177072627957767033273400054035646249294116194032597141380231809495450
92061697030016494399142190988277584837514769486510125299825183801940520117672004
69815008618834497212396877019885026305193978839493550456460179463908788738146706
76117570771260723737087908013798276896164888369728288325557614937304828056147649
35540551424499484277460294562157818345016392728403739808145556731852454497881098
31078090353806531287403422631130813673220450887226435786320034093923471350982656
30058978220686840907690004674807731987131613227786426463089188539679875639065335
87054646907057418410341784851395981864800625638521314083825618290926831970459505
97711524244467801027809221917754667758511711159477524279938158225663330114237679
17630424177574112313453492651368591903634888268873050916197211569534429443348079
80701052054477464027756544215788271144592065259518160027781799521552454229528696
4748569609435011181648374112
```

## 7.2 Text output



```
edward@edward-MS-7808:~/Desktop/libaescprintf-1.0.2$ cd bin
edward@edward-MS-7808:~/Desktop/libaescprintf-1.0.2/bin$ ls
libaescprintf
edward@edward-MS-7808:~/Desktop/libaescprintf-1.0.2/bin$ ./libaescprintf
***** -- Hello world --***** style and color nb: 01
***** -- Hello world --***** style and color nb: 02
***** -- Hello world --***** style and color nb: 03
***** -- Hello world --***** style and color nb: 04
***** -- Hello world --***** style and color nb: 05
***** -- Hello world --***** style and color nb: 06
***** -- Hello world --***** style and color nb: 07

***** -- Hello world --***** style and color nb: 09
***** -- Hello world --***** style and color nb: 10
***** -- Hello world --***** style and color nb: 11
***** -- Hello world --***** style and color nb: 12
***** -- Hello world --***** style and color nb: 13
***** -- Hello world --***** style and color nb: 14

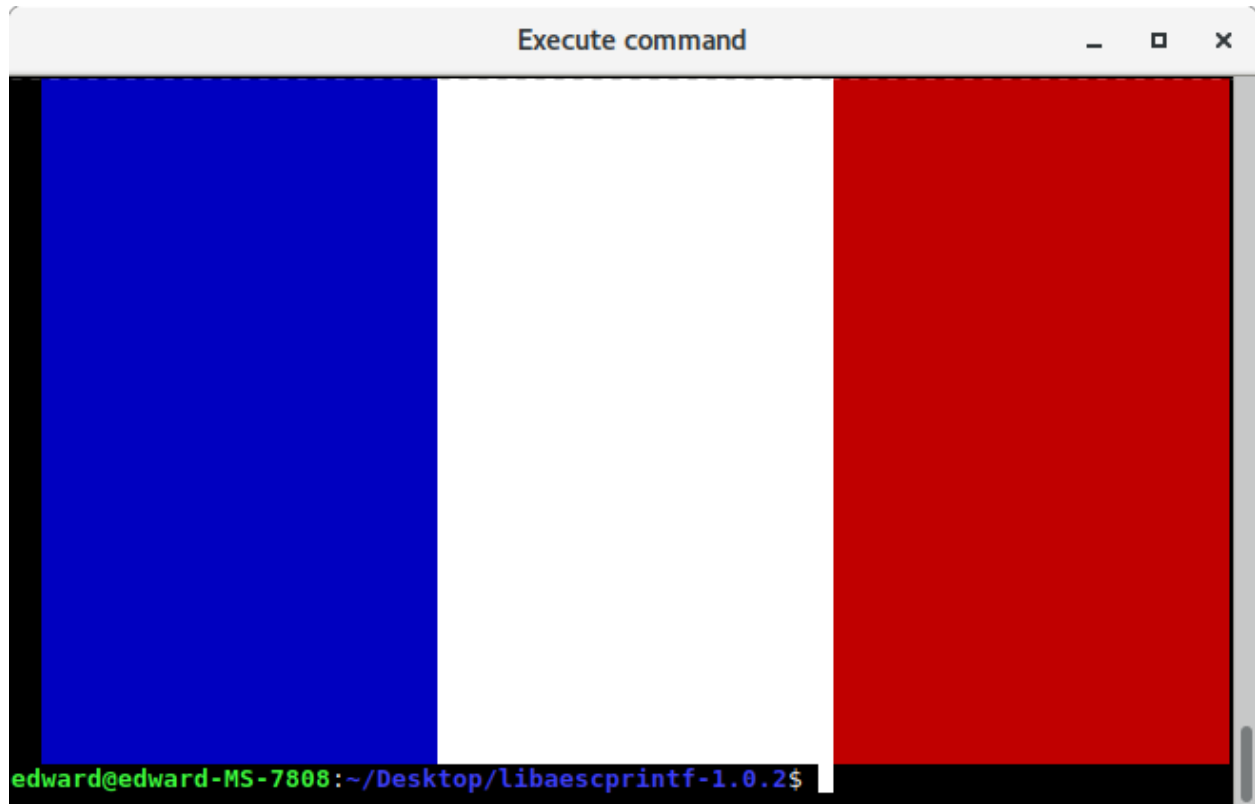
***** -- Hello world --***** style and color nb: 16

***** -- Hello world --***** style and color nb: 18

***** -- libaescprintf is great !!! --*****
edward@edward-MS-7808:~/Desktop/libaescprintf-1.0.2/bin$
```



## 7.3 Flags output





## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`



## INDEX

### C

cbg\_fprintf (C function), 13  
cbg\_printf (C function), 9  
cfg\_fprintf (C function), 13  
cfg\_printf (C function), 9  
cfgbg\_fprintf (C function), 14  
cfgbg\_printf (C function), 10

### S

s\_fprintf (C function), 13  
s\_printf (C function), 9  
scbg\_printf (C function), 10, 14  
scfg\_fprintf (C function), 14  
scfg\_printf (C function), 10  
scfgbg\_fprintf (C function), 15  
scfgbg\_printf (C function), 10