
zbarlight

Release 3.1.dev0

Jan 02, 2020

Contents

1	Installation	3
1.1	On Debian	3
1.2	On Mac OS X	3
1.3	On Windows	3
2	How To use ZbarLight	5
3	Troubleshooting	7
4	Reference	9
5	ChangeLog	11
5.1	3.1 (unreleased)	11
5.2	3.0 (2020-01-02)	11
5.3	2.3 (2019-03-21)	11
5.4	2.2 (2019-01-21)	11
5.5	2.1 (2018-06-04)	11
5.6	2.0 (2018-01-24)	12
5.7	1.2 (2017-03-09)	12
5.8	1.1.0 (2017-01-24)	12
5.9	1.0.2 (2016-08-03)	12
5.10	1.0.1 (2016-06-17)	12
5.11	1.0.0 (2015-09-25)	12
5.12	0.1.1 (2014-12-16)	12
5.13	0.1.0 (2014-12-12)	12
6	ZbarLight	13
7	Installation	15
7.1	On Debian	15
7.2	On Mac OS X	15
7.3	On Windows	15
8	How To use ZbarLight	17
9	Troubleshooting	19

zbarlight is a simple wrapper for the zbar library. For now, it can read all zbar supported codes. Contributions, suggestions and pull requests are welcome.

zbarlight is hosted on Github at <<https://github.com/Polyconseil/zbarlight/>>.

CHAPTER 1

Installation

You need to install ZBar Bar Code Reader <<http://zbar.sourceforge.net/>> and its headers before installing zbarlight.

1.1 On Debian

```
$ apt-get install libzbar0 libzbar-dev
$ pip install zbarlight # you can also use setuptools directly
```

1.2 On Mac OS X

```
$ brew install zbar
$ export LDFLAGS="-L$(brew --prefix zbar)/lib"
$ export CFLAGS="-I$(brew --prefix zbar)/include"
$ pip install zbarlight
```

1.3 On Windows

Instruction can be found on <<https://gist.github.com/Zephor5/aea563808d80f488310869b69661f330>>.

CHAPTER 2

How To use ZbarLight

```
from PIL import Image
import zbarlight

file_path = './tests/fixtures/two_qr_codes.png'
with open(file_path, 'rb') as image_file:
    image = Image.open(image_file)
    image.load()

codes = zbarlight.scan_codes(['qr'], image)
print('QR codes: %s' % codes)
```

Troubleshooting

In some case `zbarlight` will not be able to detect the 1D or 2D code in an image, one of the known cause is that the image background color is the same as the foreground color after conversion to grey scale (it's happen on images with alpha channel). You can use the `copy_image_on_background` function to add a background color on your image.

```
from PIL import Image
import zbarlight

file_path = './tests/fixtures/two_qr_codes.png'
with open(file_path, 'rb') as image_file:
    image = Image.open(image_file)
    image.load()

new_image = zbarlight.copy_image_on_background(image, color=zbarlight.WHITE) # <<<<<<<
→<<<<<<<<<< Add this line <<<<
codes = zbarlight.scan_codes(['qrcode'], new_image)
print('QR codes: %s' % codes)
```

Some other cases without known solutions are show in the `scan_codes()` tests (search for the expected failures). Any clues on these cases is welcome.

CHAPTER 4

Reference

5.1 3.1 (unreleased)

- Nothing changed yet.

5.2 3.0 (2020-01-02)

- Add Python 3.8 support
- Drop Python 2.7 support (end-of-life 2020-01-01) **breaking change**
- Drop Python 3.4 support (end-of-life 2019-03-18) **breaking change**

5.3 2.3 (2019-03-21)

- Support <https://github.com/mchehab/zbar> zbar fork (used by ArchLinux)

5.4 2.2 (2019-01-21)

- Add official support for Python 3.7
- Deprecate Python 3.4 (end-of-life 2019-03-16)

5.5 2.1 (2018-06-04)

- Allow to search for more than one kind of bar code at once
- **deprecate** `scan_codes(str, Image)` in favor of `scan_codes(list, Image)`

5.6 2.0 (2018-01-24)

- Drop deprecated `qr_code_scanner()` method
- Add helper to add background color on image

5.7 1.2 (2017-03-09)

- Only return asked symbologie

5.8 1.1.0 (2017-01-24)

- Officially support Python 3.6.
- Drop Python 2.6 support

5.9 1.0.2 (2016-08-03)

- Fix Install for `setuptools < 22.0`

5.10 1.0.1 (2016-06-17)

- Use `zest.releaser`
- Use `tox`
- Do not include tests and docs in package

5.11 1.0.0 (2015-09-25)

- Add generic `scan_codes()` function (which can scan multiple codes in the same image)
- Fix Python 2.6 tests
- Exclude tests from package

5.12 0.1.1 (2014-12-16)

- Minor fixes on Readme
- Include `requirements-dev.txt` in Manifest

5.13 0.1.0 (2014-12-12)

- First public version.

CHAPTER 6

ZbarLight

`zbarlight` is a simple wrapper for the `zbar` library. For now, it can read all `zbar` supported codes. Contributions, suggestions and pull requests are welcome.

`zbarlight` is hosted on Github at [<https://github.com/Polyconseil/zbarlight/>](https://github.com/Polyconseil/zbarlight/).

You need to install ZBar Bar Code Reader <<http://zbar.sourceforge.net/>> and its headers before installing zbarlight.

7.1 On Debian

```
$ apt-get install libzbar0 libzbar-dev
$ pip install zbarlight # you can also use setuptools directly
```

7.2 On Mac OS X

```
$ brew install zbar
$ export LDFLAGS="-L$(brew --prefix zbar)/lib"
$ export CFLAGS="-I$(brew --prefix zbar)/include"
$ pip install zbarlight
```

7.3 On Windows

Instruction can be found on <<https://gist.github.com/Zephor5/aea563808d80f488310869b69661f330>>.

How To use ZbarLight

```
from PIL import Image
import zbarlight

file_path = './tests/fixtures/two_qr_codes.png'
with open(file_path, 'rb') as image_file:
    image = Image.open(image_file)
    image.load()

codes = zbarlight.scan_codes(['qr'], image)
print('QR codes: %s' % codes)
```

Troubleshooting

In some case `zbarlight` will not be able to detect the 1D or 2D code in an image, one of the known cause is that the image background color is the same as the foreground color after conversion to grey scale (it's happen on images with alpha channel). You can use the `copy_image_on_background` function to add a background color on your image.

```
from PIL import Image
import zbarlight

file_path = './tests/fixtures/two_qr_codes.png'
with open(file_path, 'rb') as image_file:
    image = Image.open(image_file)
    image.load()

new_image = zbarlight.copy_image_on_background(image, color=zbarlight.WHITE) # <<<<<<<
→<<<<<<<<<< Add this line <<<<
codes = zbarlight.scan_codes(['qr'], new_image)
print('QR codes: %s' % codes)
```

Some other cases without known solutions are show in the `scan_codes()` tests (search for the expected failures). Any clues on these cases is welcome.