

---

# **image-diet2 Documentation**

***Release 1.0***

**Marko Samastur**

January 09, 2016



<b>1</b>	<b>Installation</b>	<b>3</b>
1.1	Distribute & Pip . . . . .	3
1.2	Get the Code . . . . .	3
1.3	Installing “dependencies” . . . . .	4
<b>2</b>	<b>Quickstart</b>	<b>5</b>
2.1	Configuration . . . . .	5
2.2	Default values . . . . .	5
2.3	DietMixin . . . . .	6
<b>3</b>	<b>Management commands</b>	<b>7</b>
3.1	check_diet_tools . . . . .	7
3.2	diet_images . . . . .	7
<b>4</b>	<b>License</b>	<b>9</b>



Release: v1.0. (*Installation*)

image-diet2 is a MIT licensed Django application for removing unnecessary bytes from image files. It optimizes images without changing their look or visual quality (“losslessly”).

It works on images in JPEG, GIF, PNG or any format with configured a processing pipeline. Integration with Django’s storage system provides a seamless integration with most thumbnailing apps.

Contents:



---

## Installation

---

This part of the documentation covers the installation of image-diet2. The first step to using any software package is getting it properly installed.

### 1.1 Distribute & Pip

Installing image-diet2 is simple with `pip`, just run this in your terminal:

```
$ pip install image-diet2
```

or, with `easy_install`:

```
$ easy_install image-diet2
```

But, you really *shouldn't* do that.

### 1.2 Get the Code

image-diet2 is developed on GitHub, where the code is *always available*.

You can either clone the public repository:

```
$ git clone git://github.com/samastur/image-diet2.git
```

Download the *tarball*:

```
$ curl -OL https://github.com/samastur/image-diet2/tarball/master
```

Or, download the *zipball*:

```
$ curl -OL https://github.com/samastur/image-diet2/zipball/master
```

Once you have a copy of the source, you can embed it in your Python package, or install it into your site-packages easily:

```
$ python setup.py install
```

## 1.3 Installing “dependencies”

image-diet2 does not have a hard dependency on any external optimisation tool, but it also does not do anything useful without any. You do need to install at least one for each image format you want to handle.

You can find a list of some in [pyimagediet’s documentation](#).



---

## Quickstart

---

### 2.1 Configuration

After you have installed image-diet package and external compression tools, you need to configure it.

First, you need to point image-diet to your configuration file which you do by adding `DIET_CONFIG` setting to your project's settings. Its value is the absolute path to the configuration file you want to use.

Configuration is stored in YAML format and is described in [pyimagediet's documentation](#) (image-diet2 uses pyimagediet for actual processing).

An additional value you can set in image-diet2's configuration file is `tmpdir` that should point to directory where temporary files will be created. Its default value is `/tmp`.

If you are using filesystem (Django's default) as storage and would like to process files with image-diet2 everywhere it is used, then add to settings:

```
DEFAULT_FILE_STORAGE = 'image_diet.storage.DietStorage'
```

If you are using some other backend class that you would like to augment with `DietStorage`, then set `DIET_STORAGE` setting to that storage class.

### 2.2 Default values

image-diet2 already comes with some default values so you do not have to know or type everything. It is enough to provide only changes you want to make and they will either replace previous ones or be added if they are new.

Default configuration file:

```
# Commands to be executed (label: path)
commands:
  optipng: optipng
  advpng: advpng
  pngcrush: pngcrush
  jpegoptim: jpegoptim
  jpegtran: jpegtran
  gifsicle: gifsicle

# Parameters for commands (label: parameters)
# Use same labels as in command section.
parameters:
```

```
optipng: -force -o7 '{file}'
advpng: -z4 '{file}s'
pngcrush: -rem gAMA -rem alla -rem cHRM -rem iCCP -rem sRGB
          -rem time '{file}' '{output_file}'

jpegoptim: -f --strip-all '{file}'
jpegtran: -copy none -progressive -optimize -outfile '{output_file}' '{file}'

gifsicle: -O2 '{file}' > '{output_file}'

# Pipelines for each file type. Order of labels specifies order of execution
# Use same labels as in command section.
pipelines: {}

# Uncomment and set if you want to backup original image
# backup: orig

# By default pyimagediet returns smallest file it can make even if that is
# the original. If you don't want that (for example to reliably measure
# effectiveness of tools and their parameters) then uncomment next line.
keep_processed: true

# Directory for creating temporary files. Defaults to /tmp when not set.
# tmpdir: /tmp
```

## 2.3 DietMixin

In case your project uses different storage backends or want to use compression only on non-default storage backend then you should use `image_diet.storage.DietMixin` mixin.

---

## Management commands

---

image-diet2 comes with management commands to make your life a bit easier.

### 3.1 check\_diet\_tools

This command will check system for most common compression tools and print paths of those found in format ready for inclusion in YAML configuration file.

You can also use command line utility *diet* that comes with pyimagediet which was installed together with image-diet2. You can read about how to use it in pyimagediet's documentation

### 3.2 diet\_images

This command will traverse provided list of directories and compress all files with matching pipeline according to image-diet2's configuration.



---

### License

---

The MIT License (MIT)

Copyright (c) 2015 Marko Samastur

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.