
src Documentation

Release 1.8.8

Author

Feb 20, 2019

1	Installation	3
2	Getting started	5
3	Features	11
4	Indices and tables	97
	Python Module Index	99

psd-tools is a Python package for working with Adobe Photoshop PSD files as described in [specification](#).

CHAPTER 1

Installation

Use *pip* to install the package:

```
pip install psd-tools
```

Note: In order to extract images from 32bit PSD files PIL/Pillow must be built with LITTLECMS or LITTLECMS2 support.

For complete layer image composition functionality, also install NumPy/SciPy. This will be only necessary when the PSD files are saved without maximized compatibility and the image contains gradient fill:

```
pip install numpy scipy
```


CHAPTER 2

Getting started

```
from psd_tools import PSDImage

psd = PSDImage.open('example.psd')
psd.compose().save('example.png')

for layer in psd:
    print(layer)
```

Check out the *Usage* documentation for more examples.

2.1 Usage

2.1.1 Command line

The package provides command line tools to handle a PSD document:

```
psd-tools export <input_file> <output_file> [options]
psd-tools show <input_file> [options]
psd-tools debug <input_file> [options]
psd-tools -h | --help
psd-tools --version
```

Example:

```
psd-tools show example.psd # Show the file content
psd-tools export example.psd example.png # Export as PNG
psd-tools export example.psd[0] example-0.png # Export layer as PNG
```

2.1.2 Working with PSD document

`psd_tools.api` package provides the user-friendly API to work with PSD files. *PSDImage* represents a PSD file.

Open an image:

```
from psd_tools import PSDImage
psd = PSDImage.open('my_image.psd')
```

Most of the data structure in the `psd-tools` supports pretty printing in IPython environment.

```
In [1]: PSDImage.open('example.psd')
Out[1]:
PSDImage(mode=RGB size=101x55 depth=8 channels=3)
  [0] PixelLayer('Background' size=101x55)
  [1] PixelLayer('Layer 1' size=85x46)
```

Internal layers are accessible by iterator or indexing:

```
for layer in psd:
    print(layer)
    if layer.is_group():
        for child in layer:
            print(child)

child = psd[0][0]
```

Note: The iteration order is from background to foreground, which is reversed from version prior to 1.7.x. Use `reversed(list(psd))` to iterate from foreground to background.

The opened file can be saved:

```
psd.save('output.psd')
```

2.1.3 Working with Layers

There are various layer kinds in Photoshop.

The most basic layer type is *PixelLayer*:

```
print(layer.name)
layer.kind == 'pixel'
```

Some of the layer attributes are editable, such as a layer name:

```
layer.name = 'Updated layer 1'
```

Note: Currently, the package does not support adding or removing of a layer.

Group has internal layers:

```
for layer in group:
    print(layer)

first_layer = group[0]
```

TypeLayer is a layer with texts:

```
print(layer.text)
```

ShapeLayer draws a vector shape, and the shape information is stored in *vector_mask* and *origination* property. Other layers can also have shape information as a mask:

```
print(layer.vector_mask)
for shape in layer.origination:
    print(shape)
```

SmartObjectLayer embeds or links an external file for non-destructive editing. The file content is accessible via *smart_object* property:

```
import io
if layer.smart_object.filetype in ('jpg', 'png'):
    image = Image.open(io.BytesIO(layer.smart_object.data))
```

SolidColorFill, *PatternFill*, and *GradientFill* are fill layers that paint the entire region if there is no associated mask. Sub-classes of *AdjustmentLayer* represents layer adjustment applied to the composed image. See *Adjustment layers*.

2.1.4 Exporting data to PIL

Export the entire document as `PIL.Image`:

```
image = psd.compose()
image.save('exported.png')
```

Note that above `compose()` might return `None` if the PSD document has no visible pixel.

Export a single layer including masks and clipping layers:

```
image = layer.compose()
```

Export layer, mask, or clipping layers separately without composition:

```
image = layer.topil()
mask = layer.mask.topil()

from psd_tools import compose
clip_image = compose(layer.clip_layers)
```

2.2 Migrating from version 1.7 to 1.8

There are major API changes in version 1.8.x.

Note: In version 1.8.0 - 1.8.7, the package name was *psd_tools2*.

2.2.1 PSDImage

File open method is changed from *load* to *open()*.

version 1.7.x:

```
psd = PSDImage.load(filename)
with open(filename, 'rb') as f:
    psd = PSDImage.from_stream(f)
```

version 1.8.x:

```
psd = PSDImage.open(filename)
with open(filename, 'rb') as f:
    psd = PSDImage.open(f)
```

2.2.2 Layers

Children of PSDImage or Group is directly accessible by iterator or indexing.

version 1.7.x:

```
for layer in group.layers:
    print(layer)

first_child = group.layers[0]
```

version 1.8.x:

```
for layer in group:
    print(layer)

first_child = group[0]
```

In version 1.8.x, the order of layers is reversed to reflect that the index should not change when a new layer is added on top.

2.2.3 PIL export

Primary PIL export method is now *compose()*.

version 1.7.x:

```
image = psd.as_PIL()

layer_image = compose(layer)
raw_layer_image = layer.as_PIL()
```

version 1.8.x:

```
image = psd.compose()

layer_image = layer.compose()
raw_layer_image = layer.topil()
```

2.2.4 Low-level data structure

Data structures are completely rewritten to support writing functionality. See `psd_tools.psd` subpackage. version 1.7.x:

```
psd.decoded_data
```

version 1.8.x:

```
psd._record
```

2.3 Contributing

Development happens at github: [bug tracker](#). Feel free to submit [bug reports](#) or pull requests. Attaching an erroneous PSD file makes the debugging process faster. Such PSD file might be added to the test suite.

The license is MIT.

2.3.1 Package design

The package consists of two major subpackages:

1. **`psd_tools.psd`: subpackage that reads/writes low-level binary** structure of the PSD/PSB file. The core data structures are built around `attrs` class that all implement `read` and `write` methods. Each data object tries to resemble the structure described in the [specification](#). Although documented, the [specification](#) is far from complete and some are even inaccurate. When `psd-tools` finds unknown data structure, the package keeps such data as `bytes` in the parsed result.
2. **`psd_tools.api`: User-facing API that implements various** easy-to-use methods that manipulate low-level `psd_tools.psd` data structures.

2.3.2 Testing

In order to run tests, make sure PIL/Pillow is built with LittleCMS or LittleCMS2 support, install `tox` and type:

```
tox
```

from the source checkout. Or, it is a good idea to install and run `detox` for parallel execution:

```
detox
```

2.3.3 Documentation

Install Sphinx to generate documents:

```
pip install sphinx sphinx_rtd_theme
```

Once installed, use *Makefile*:

```
make docs
```

2.3.4 Acknowledgments

Great thanks to all the contributors.

Supported:

- Read and write of the low-level PSD/PSB file structure;
- Raw layer image export;
- ICC profile handling for sRGB images.

Limited support:

- Composition of basic pixel-based layers by normal blending;
- Composition of fill layer effects;
- Editing of some layer attributes such as layer name.

Not supported:

- Editing of layer structure, such as adding or removing a layer.
- Blending modes other than normal;
- Composition of layer effects;
- Drawing of bezier curves;
- Font rendering.

3.1 psd_tools

See *Usage* for examples.

3.1.1 PSDImage

```
class psd_tools.PSDImage(data)  
    Photoshop PSD/PSB file object.
```

Example:

```
from psd_tools import PSDImage

psd = PSDImage.open('example.psd')
image = psd.topil()

for layer in psd:
    if layer.has_pixels():
        layer_image = layer.topil()
```

bbox

Minimal bounding box that contains all the visible layers.

Use `viewbox` to get viewport bounding box. When the psd is empty, `bbox` is equal to the canvas bounding box.

Returns (left, top, right, bottom) *tuple*.

bottom

Bottom coordinate.

Returns *int*

channels

Number of color channels.

Returns *int*

color_mode

Document color mode, such as 'RGB' or 'GRAYSCALE'. See [ColorMode](#).

Returns *str*

compose (*force=False, bbox=None, **kwargs*)

Compose the PSD image.

See `compose()` for available extra arguments.

Parameters `bbox` – Viewport tuple (left, top, right, bottom).

Returns `PIL.Image`, or `None` if there is no pixel.

depth

Pixel depth bits.

Returns *int*

descendants (*include_clip=True*)

Return a generator to iterate over all descendant layers.

Example:

```
# Iterate over all layers
for layer in psd.descendants():
    print(layer)

# Iterate over all layers in reverse order
for layer in reversed(list(psd.descendants())):
    print(layer)
```

Parameters `include_clip` – include clipping layers.

classmethod frompil (*image*, *compression=<Compression.PACK_BITS: 1>*)

Create a new PSD document from PIL Image.

Parameters

- **image** – PIL Image object.
- **compression** – ImageData compression option. See *Compression*.

Returns A PSDImage object.

has_preview ()

Returns if the document has real merged data. When True, *topil()* returns pre-composed data.

has_thumbnail ()

True if the PSDImage has a thumbnail resource.

height

Document height.

Returns *int*

image_resources

Document image resources.

Returns ImageResouces.

is_group ()

Return True if the layer is a group.

Returns *bool*

is_visible ()

Returns visibility of the element.

Returns *bool*

kind

Kind.

Returns *'psdimage'*

left

Left coordinate.

Returns *0*

name

Element name.

Returns *'Root'*

classmethod new (*mode*, *size*, *color=0*, *depth=8*, ***kwargs*)

Create a new PSD document.

Parameters

- **mode** – The color mode to use for the new image.
- **size** – A tuple containing (width, height) in pixels.
- **color** – What color to use for the image. Default is black.

Returns A PSDImage object.

offset

(left, top) tuple.

Returns *tuple*

classmethod open (*fp*)

Open a PSD document.

Parameters **fp** – filename or file-like object.

Returns A PSDImage object.

parent

Parent of this layer.

right

Right coordinate.

Returns *int*

save (*fp, mode='wb'*)

Save the PSD file.

Parameters

- **fp** – filename or file-like object.
- **mode** – file open mode, default 'wb'.

size

(width, height) tuple.

Returns *tuple*

tagged_blocks

Document tagged blocks.

Returns *TaggedBlocks* or *None*.

thumbnail ()

Returns a thumbnail image in PIL.Image. When the file does not contain an embedded thumbnail image, returns *None*.

top

Top coordinate.

Returns *0*

topil ()

Get PIL Image.

Returns *PIL. Image*, or *None* if the composed image is not available.

version

Document version. PSD file is 1, and PSB file is 2.

Returns *int*

viewbox

Return bounding box of the viewport.

Returns (left, top, right, bottom) *tuple*.

visible

Visibility.

Returns *True*

width

Document width.

Returns *int*

3.1.2 compose

`psd_tools.compose` (*layers, bbox=None, layer_filter=None, color=None*)

Compose layers to a single `PIL.Image`. If the layers do not have visible pixels, the function returns *None*.

Example:

```
image = compose([layer1, layer2])
```

In order to skip some layers, pass *layer_filter* function which should take *layer* as an argument and return *True* to keep the layer or return *False* to skip:

```
image = compose(
    layers,
    layer_filter=lambda x: x.is_visible() and x.kind == 'type'
)
```

By default, visible layers are composed.

Note: This function is experimental and does not guarantee Photoshop-quality rendering.

Currently the following are ignored:

- Adjustments layers
- Layer effects
- Blending mode (all blending modes become normal)

Shape drawing is inaccurate if the PSD file is not saved with maximum compatibility.

Parameters

- **layers** – a layer, or an iterable of layers.
- **bbox** – (left, top, bottom, right) tuple that specifies a region to compose. By default, all the visible area is composed. The origin is at the top-left corner of the PSD document.
- **layer_filter** – a callable that takes a layer and returns *bool*.
- **color** – background color in *int* or *tuple*.

Returns `PIL.Image` or *None*.

3.2 psd_tools.api.adjustments

Adjustment and fill layers.

Example:

```
if layer.kind == 'brightnesscontrast':
    print(layer.brightness)

if layer.kind == 'gradient-fill':
    print(layer.gradient_kind)
```

3.2.1 Fill layers

Fill layers are similar to *ShapeLayer* except that the layer might not have an associated vector mask. The layer therefore expands the entire canvas of the PSD document.

Fill layers all inherit from `FillLayer`.

Example:

```
if isinstance(layer, psd_tools.layers.FillLayer):
    image = layer.compose()
```

class `psd_tools.api.adjustments.SolidColorFill(*args)`
Solid color fill.

bbox
(left, top, right, bottom) tuple.

blend_mode
Blend mode of this layer. Writable.

Example:

```
from psd_tools.constants import BlendMode
if layer.blend_mode == BlendMode.NORMAL:
    layer.blend_mode = BlendMode.SCREEN
```

Returns *BlendMode*.

bottom
Bottom coordinate.

Returns int

clip_layers
Clip layers associated with this layer.

To compose clipping layers:

```
from psd_tools import compose
clip_mask = compose(layer.clip_layers)
```

Returns list of layers

compose(*args, **kwargs)
Compose layer and masks (mask, vector mask, and clipping layers).

Returns `PIL.Image`, or `None` if the layer has no pixel.

data
Color in Descriptor(RGB).

effects
Layer effects.

Returns *Effects*

has_clip_layers()
Returns True if the layer has associated clipping.

Returns *bool*

has_effects ()
Returns True if the layer has effects.
Returns *bool*

has_mask ()
Returns True if the layer has a mask.
Returns *bool*

has_origination ()
Returns True if the layer has live shape properties.
Returns *bool*

has_pixels ()
Returns True if the layer has associated pixels. When this is True, `topil` method returns `PIL.Image`.
Returns *bool*

has_vector_mask ()
Returns True if the layer has a vector mask.
Returns *bool*

height
Height of the layer.
Returns *int*

is_group ()
Return True if the layer is a group.
Returns *bool*

is_visible ()
Layer visibility. Takes group visibility in account.
Returns *bool*

kind
Kind of this layer, either of group, pixel, shape, type, smartobject, or psdimage.
Returns *str*

layer_id
Layer ID.
Returns *int* layer id. if the layer is not assigned an id, -1.

left
Left coordinate. Writable.
Returns *int*

mask
Returns mask associated with this layer.
Returns *Mask* or *None*

name
Layer name. Writable.
Returns *str*

offset
(left, top) tuple. Writable.

Returns *tuple*

opacity

Opacity of this layer in [0, 255] range. Writable.

Returns *int*

origination

Property for a list of live shapes or a line.

Some of the vector masks have associated live shape properties, that are Photoshop feature to handle primitive shapes such as a rectangle, an ellipse, or a line. Vector masks without live shape properties are plain path objects.

Returns List of *Invalidated, Rectangle, RoundedRectangle, Ellipse, or Line*.

parent

Parent of this layer.

right

Right coordinate.

Returns *int*

size

(width, height) tuple.

Returns *tuple*

top

Top coordinate. Writable.

Returns *int*

topil ()

Get PIL Image of the layer.

Returns *PIL. Image*, or *None* if the layer has no pixels.

vector_mask

Returns vector mask associated with this layer.

Returns *VectorMask* or *None*

visible

Layer visibility. Doesn't take group visibility in account. Writable.

Returns *bool*

width

Width of the layer.

Returns *int*

class `psd_tools.api.adjustments.PatternFill (*args)`

Pattern fill.

bbox

(left, top, right, bottom) tuple.

blend_mode

Blend mode of this layer. Writable.

Example:

```

from psd_tools.constants import BlendMode
if layer.blend_mode == BlendMode.NORMAL:
    layer.blend_mode = BlendMode.SCREEN

```

Returns *BlendMode*.

bottom

Bottom coordinate.

Returns *int*

clip_layers

Clip layers associated with this layer.

To compose clipping layers:

```

from psd_tools import compose
clip_mask = compose(layer.clip_layers)

```

Returns *list of layers*

compose (*args, **kwargs)

Compose layer and masks (mask, vector mask, and clipping layers).

Returns *PIL. Image*, or *None* if the layer has no pixel.

data

Pattern in Descriptor(PATTERN).

effects

Layer effects.

Returns *Effects*

has_clip_layers ()

Returns True if the layer has associated clipping.

Returns *bool*

has_effects ()

Returns True if the layer has effects.

Returns *bool*

has_mask ()

Returns True if the layer has a mask.

Returns *bool*

has_origination ()

Returns True if the layer has live shape properties.

Returns *bool*

has_pixels ()

Returns True if the layer has associated pixels. When this is True, *topil* method returns *PIL. Image*.

Returns *bool*

has_vector_mask ()

Returns True if the layer has a vector mask.

Returns *bool*

height

Height of the layer.

Returns *int*

is_group()

Return True if the layer is a group.

Returns *bool*

is_visible()

Layer visibility. Takes group visibility in account.

Returns *bool*

kind

Kind of this layer, either of group, pixel, shape, type, smartobject, or psdimage.

Returns *str*

layer_id

Layer ID.

Returns *int* layer id. if the layer is not assigned an id, -1.

left

Left coordinate. Writable.

Returns *int*

mask

Returns mask associated with this layer.

Returns *Mask* or *None*

name

Layer name. Writable.

Returns *str*

offset

(left, top) tuple. Writable.

Returns *tuple*

opacity

Opacity of this layer in [0, 255] range. Writable.

Returns *int*

origination

Property for a list of live shapes or a line.

Some of the vector masks have associated live shape properties, that are Photoshop feature to handle primitive shapes such as a rectangle, an ellipse, or a line. Vector masks without live shape properties are plain path objects.

Returns List of *Invalidated*, *Rectangle*, *RoundedRectangle*, *Ellipse*, or *Line*.

parent

Parent of this layer.

right

Right coordinate.

Returns int

size

(width, height) tuple.

Returns *tuple*

top

Top coordinate. Writable.

Returns int

topil()

Get PIL Image of the layer.

Returns `PIL.Image`, or *None* if the layer has no pixels.

vector_mask

Returns vector mask associated with this layer.

Returns *VectorMask* or *None*

visible

Layer visibility. Doesn't take group visibility in account. Writable.

Returns *bool*

width

Width of the layer.

Returns int

class `psd_tools.api.adjustments.GradientFill(*args)`

Gradient fill.

bbox

(left, top, right, bottom) tuple.

blend_mode

Blend mode of this layer. Writable.

Example:

```
from psd_tools.constants import BlendMode
if layer.blend_mode == BlendMode.NORMAL:
    layer.blend_mode = BlendMode.SCREEN
```

Returns *BlendMode*.

bottom

Bottom coordinate.

Returns int

clip_layers

Clip layers associated with this layer.

To compose clipping layers:

```
from psd_tools import compose
clip_mask = compose(layer.clip_layers)
```

Returns list of layers

compose (*args, **kwargs)

Compose layer and masks (mask, vector mask, and clipping layers).

Returns `PIL.Image`, or `None` if the layer has no pixel.

data

Gradient in Descriptor(`GRADIENT`).

effects

Layer effects.

Returns `Effects`

gradient_kind

Kind of the gradient, one of the following:

- *linear*
- *radial*
- *angle*
- *reflected*
- *diamond*

has_clip_layers ()

Returns True if the layer has associated clipping.

Returns `bool`

has_effects ()

Returns True if the layer has effects.

Returns `bool`

has_mask ()

Returns True if the layer has a mask.

Returns `bool`

has_origination ()

Returns True if the layer has live shape properties.

Returns `bool`

has_pixels ()

Returns True if the layer has associated pixels. When this is True, `topil` method returns `PIL.Image`.

Returns `bool`

has_vector_mask ()

Returns True if the layer has a vector mask.

Returns `bool`

height

Height of the layer.

Returns `int`

is_group ()

Return True if the layer is a group.

Returns `bool`

is_visible()

Layer visibility. Takes group visibility in account.

Returns *bool*

kind

Kind of this layer, either of group, pixel, shape, type, smartobject, or psdimage.

Returns *str*

layer_id

Layer ID.

Returns int layer id. if the layer is not assigned an id, -1.

left

Left coordinate. Writable.

Returns int

mask

Returns mask associated with this layer.

Returns *Mask* or *None*

name

Layer name. Writable.

Returns *str*

offset

(left, top) tuple. Writable.

Returns *tuple*

opacity

Opacity of this layer in [0, 255] range. Writable.

Returns int

origination

Property for a list of live shapes or a line.

Some of the vector masks have associated live shape properties, that are Photoshop feature to handle primitive shapes such as a rectangle, an ellipse, or a line. Vector masks without live shape properties are plain path objects.

Returns List of *Invalidated*, *Rectangle*, *RoundedRectangle*, *Ellipse*, or *Line*.

parent

Parent of this layer.

right

Right coordinate.

Returns int

size

(width, height) tuple.

Returns *tuple*

top

Top coordinate. Writable.

Returns int

topil()

Get PIL Image of the layer.

Returns `PIL.Image`, or `None` if the layer has no pixels.**vector_mask**

Returns vector mask associated with this layer.

Returns `VectorMask` or `None`**visible**

Layer visibility. Doesn't take group visibility in account. Writable.

Returns `bool`**width**

Width of the layer.

Returns `int`

3.2.2 Adjustment layers

Adjustment layers apply image filtering to the composed result. All adjustment layers inherit from `AdjustmentLayer`. Adjustment layers do not have pixels, and currently ignored in `compose`. Attempts to call `topil` on adjustment layers always return `None`.

Just as any other layer, adjustment layers might have an associated mask or vector mask. Adjustment can appear in other layers' clipping layers.

Example:

```
if isinstance(layer, psd_tools.layers.AdjustmentLayer):
    print(layer.kind)
```

class `psd_tools.api.adjustments.BrightnessContrast(*args)`

Brightness and contrast adjustment.

automatic**brightness****contrast****lab****mean****use_legacy****vrsn****class** `psd_tools.api.adjustments.Curves(*args)`

Curves adjustment.

data

Raw data.

Returns `Curves`**extra****class** `psd_tools.api.adjustments.Exposure(*args)`

Exposure adjustment.

exposure

Exposure.

Returns *float*

gamma

Gamma.

Returns *float*

offset

Offset.

Returns *float*

class `psd_tools.api.adjustments.Levels(*args)`

Levels adjustment.

Levels contain a list of `LevelRecord`.

data

List of level records. The first record is the master.

Returns `Levels`.

master

Master record.

class `psd_tools.api.adjustments.Vibrance(*args)`

Vibrance adjustment.

saturation

Saturation.

Returns *int*

vibrance

Vibrance.

Returns *int*

class `psd_tools.api.adjustments.HueSaturation(*args)`

Hue/Saturation adjustment.

HueSaturation contains a list of data.

colorization

Colorization.

Returns *tuple*

data

List of Hue/Saturation records.

Returns *list*

enable_colorization

Enable colorization.

Returns *int*

master

Master record.

Returns *tuple*

class psd_tools.api.adjustments.**ColorBalance**(*args)
Color balance adjustment.

highlights
Highlights.

Returns *tuple*

luminosity
Luminosity.

Returns *int*

midtone
Mid-tones.

Returns *tuple*

shadows
Shadows.

Returns *tuple*

class psd_tools.api.adjustments.**BlackAndWhite**(*args)
Black and white adjustment.

blue

cyan

green

magenta

preset_file_name

preset_kind

red

tint_color

use_tint

yellow

class psd_tools.api.adjustments.**PhotoFilter**(*args)
Photo filter adjustment.

color_components

color_space

density

luminosity

xyz

xyz.

Returns *bool*

class psd_tools.api.adjustments.**ChannelMixer**(*args)
Channel mixer adjustment.

data

monochrome

```
class psd_tools.api.adjustments.ColorLookup(*args)
    Color lookup adjustment.

class psd_tools.api.adjustments.Posterize(*args)
    Posterize adjustment.

    posterize
        Posterize value.

        Returns int

class psd_tools.api.adjustments.Threshold(*args)
    Threshold adjustment.

    threshold
        Threshold value.

        Returns int

class psd_tools.api.adjustments.SelectiveColor(*args)
    Selective color adjustment.

    data

    method

class psd_tools.api.adjustments.GradientMap(*args)
    Gradient map adjustment.

    color_model

    color_stops

    dithered

    expansion

    gradient_name

    interpolation
        Interpolation between 0.0 and 1.0.

    length

    max_color

    min_color

    mode

    random_seed

    reversed

    roughness

    show_transparency

    transparency_stops

    use_vector_color
```

3.3 psd_tools.api.effects

Effects module.

class `psd_tools.api.effects.Effects` (*layer*)

List-like effects.

enabled

Whether if all the effects are enabled.

Return type `bool`

scale

Scale value.

3.3.1 DropShadow

class `psd_tools.api.effects.DropShadow` (*value, image_resources*)

angle

Angle value.

anti_aliased

Anti-aliased.

blend_mode

Effect blending mode.

choke

Choke level.

color

Color.

contour

Contour configuration.

distance

Distance.

enabled

Whether if the effect is enabled.

layer_knocks_out

Layers are knocking out.

noise

Noise level.

opacity

Layer effect opacity in percentage.

present

Whether if the effect is present in Photoshop UI.

shown

Whether if the effect is shown in dialog.

size

Size in pixels.

use_global_light

Using global light.

3.3.2 InnerShadow

class `psd_tools.api.effects.InnerShadow` (*value, image_resources*)

angle
Angle value.

anti_aliased
Angi-aliased.

blend_mode
Effect blending mode.

choke
Choke level.

color
Color.

contour
Contour configuration.

distance
Distance.

enabled
Whether if the effect is enabled.

noise
Noise level.

opacity
Layer effect opacity in percentage.

present
Whether if the effect is present in Photoshop UI.

shown
Whether if the effect is shown in dialog.

size
Size in pixels.

use_global_light
Using global light.

3.3.3 OuterGlow

class `psd_tools.api.effects.OuterGlow` (*value, image_resources*)

anti_aliased
Angi-aliased.

blend_mode
Effect blending mode.

choke
Choke level.

color
Color.

contour
Contour configuration.

enabled
Whether if the effect is enabled.

glow_type
Glow type. 'softer' or 'precise'.

gradient
Gradient configuration.

noise
Noise level.

opacity
Layer effect opacity in percentage.

present
Whether if the effect is present in Photoshop UI.

quality_jitter
Quality jitter

quality_range
Quality range.

shown
Whether if the effect is shown in dialog.

size
Size in pixels.

3.3.4 InnerGlow

class `psd_tools.api.effects.InnerGlow` (*value, image_resources*)

anti_aliased
Anti-aliased.

blend_mode
Effect blending mode.

choke
Choke level.

color
Color.

contour
Contour configuration.

enabled
Whether if the effect is enabled.

glow_source
Elements source, 'edge' or 'center'.

glow_type
Glow type. 'softer' or 'precise'.

gradient
Gradient configuration.

noise
Noise level.

opacity
Layer effect opacity in percentage.

present
Whether if the effect is present in Photoshop UI.

quality_jitter
Quality jitter

quality_range
Quality range.

shown
Whether if the effect is shown in dialog.

size
Size in pixels.

3.3.5 ColorOverlay

class `psd_tools.api.effects.ColorOverlay` (*value, image_resources*)

blend_mode
Effect blending mode.

color
Color.

enabled
Whether if the effect is enabled.

opacity
Layer effect opacity in percentage.

present
Whether if the effect is present in Photoshop UI.

shown
Whether if the effect is shown in dialog.

3.3.6 GradientOverlay

class `psd_tools.api.effects.GradientOverlay` (*value, image_resources*)

aligned
Aligned.

angle
Angle value.

blend_mode

Effect blending mode.

dithered

Dither flag.

enabled

Whether if the effect is enabled.

gradient

Gradient configuration.

offset

Offset value.

opacity

Layer effect opacity in percentage.

present

Whether if the effect is present in Photoshop UI.

reversed

Reverse flag.

scale

Scale value.

shown

Whether if the effect is shown in dialog.

type

Gradient type, one of *linear*, *radial*, *angle*, *reflected*, or *diamond*.

3.3.7 PatternOverlay

class `psd_tools.api.effects.PatternOverlay` (*value*, *image_resources*)

aligned

Aligned.

blend_mode

Effect blending mode.

enabled

Whether if the effect is enabled.

opacity

Layer effect opacity in percentage.

pattern

Pattern config.

phase

Phase value in Point.

present

Whether if the effect is present in Photoshop UI.

scale

Scale value.

shown
Whether if the effect is shown in dialog.

3.3.8 Stroke

class `psd_tools.api.effects.Stroke` (*value, image_resources*)

blend_mode
Effect blending mode.

color
Color.

enabled
Whether if the effect is enabled.

fill_type
Fill type, solid-color, gradient, or pattern.

gradient
Gradient configuration.

opacity
Layer effect opacity in percentage.

overprint
Overprint flag.

pattern
Pattern config.

position
Position of the stroke, *inner*, *outer*, or *center*.

present
Whether if the effect is present in Photoshop UI.

shown
Whether if the effect is shown in dialog.

size
Size value.

3.3.9 BevelEmboss

class `psd_tools.api.effects.BevelEmboss` (*value, image_resources*)

altitude
Altitude value.

angle
Angle value.

anti_aliased
Anti-aliased.

bevel_style
Bevel style, one of *outer*, *inner*, *emboss*, *pillow-emboss*, or *stroke-emboss*.

bevel_type
Bevel type, one of *smooth*, *chiesel-hard*, *chiesel-soft*.

contour
Contour configuration.

depth
Depth value.

direction
Direction, either *in* or *out*.

enabled
Whether if the effect is enabled.

highlight_color
Highlight color value.

highlight_mode
Highlight blending mode.

highlight_opacity
Highlight opacity value.

opacity
Layer effect opacity in percentage.

present
Whether if the effect is present in Photoshop UI.

shadow_color
Shadow color value.

shadow_mode
Shadow blending mode.

shadow_opacity
Shadow opacity value.

shown
Whether if the effect is shown in dialog.

size
Size value in pixel.

soften
Soften value.

use_global_light
Using global light.

use_shape
Using shape.

use_texture
Using texture.

3.3.10 Satin

class `psd_tools.api.effects.Satin` (*value*, *image_resources*)
Satin effect

angle	Angle value.
anti_aliased	Anti-aliased.
blend_mode	Effect blending mode.
color	Color.
contour	Contour configuration.
distance	Distance value.
enabled	Whether if the effect is enabled.
inverted	Inverted.
opacity	Layer effect opacity in percentage.
present	Whether if the effect is present in Photoshop UI.
shown	Whether if the effect is shown in dialog.
size	Size value in pixel.

3.4 psd_tools.api.layers

Layer module.

3.4.1 Group

class `psd_tools.api.layers.Group(*args)`
Group of layers.

Example:

```
group = psd[1]
for layer in group:
    if layer.kind == 'pixel':
        print(layer.name)
```

bbox
(left, top, right, bottom) tuple.

blend_mode
Blend mode of this layer. Writable.

Example:

```
from psd_tools.constants import BlendMode
if layer.blend_mode == BlendMode.NORMAL:
    layer.blend_mode = BlendMode.SCREEN
```

Returns *BlendMode*.

clip_layers

Clip layers associated with this layer.

To compose clipping layers:

```
from psd_tools import compose
clip_mask = compose(layer.clip_layers)
```

Returns list of layers

compose()

Compose layer and masks (mask, vector mask, and clipping layers).

Returns PIL Image object, or None if the layer has no pixels.

descendants (*include_clip=True*)

Return a generator to iterate over all descendant layers.

Example:

```
# Iterate over all layers
for layer in psd.descendants():
    print(layer)

# Iterate over all layers in reverse order
for layer in reversed(list(psd.descendants())):
    print(layer)
```

Parameters **include_clip** – include clipping layers.

effects

Layer effects.

Returns *Effects*

has_clip_layers()

Returns True if the layer has associated clipping.

Returns *bool*

has_effects()

Returns True if the layer has effects.

Returns *bool*

has_mask()

Returns True if the layer has a mask.

Returns *bool*

has_origination()

Returns True if the layer has live shape properties.

Returns *bool*

has_pixels ()

Returns True if the layer has associated pixels. When this is True, `topil` method returns `PIL.Image`.

Returns *bool*

has_vector_mask ()

Returns True if the layer has a vector mask.

Returns *bool*

height

Height of the layer.

Returns *int*

is_group ()

Return True if the layer is a group.

Returns *bool*

is_visible ()

Layer visibility. Takes group visibility in account.

Returns *bool*

kind

Kind of this layer, either of group, pixel, shape, type, smartobject, or psdimage.

Returns *str*

layer_id

Layer ID.

Returns *int* layer id. if the layer is not assigned an id, -1.

mask

Returns mask associated with this layer.

Returns *Mask* or *None*

name

Layer name. Writable.

Returns *str*

offset

(left, top) tuple. Writable.

Returns *tuple*

opacity

Opacity of this layer in [0, 255] range. Writable.

Returns *int*

origination

Property for a list of live shapes or a line.

Some of the vector masks have associated live shape properties, that are Photoshop feature to handle primitive shapes such as a rectangle, an ellipse, or a line. Vector masks without live shape properties are plain path objects.

Returns List of *Invalidated*, *Rectangle*, *RoundedRectangle*, *Ellipse*, or *Line*.

parent

Parent of this layer.

size
(width, height) tuple.
Returns *tuple*

topil()
Get PIL Image of the layer.
Returns `PIL.Image`, or *None* if the layer has no pixels.

vector_mask
Returns vector mask associated with this layer.
Returns *VectorMask* or *None*

visible
Layer visibility. Doesn't take group visibility in account. Writable.
Returns *bool*

width
Width of the layer.
Returns *int*

3.4.2 PixelLayer

class `psd_tools.api.layers.PixelLayer` (*psd, record, channels, parent*)
Layer that has rasterized image in pixels.

Example:

```
assert layer.kind == 'pixel':
image = layer.topil()
image.save('layer.png')

composed_image = layer.compose()
composed_image.save('composed-layer.png')
```

bbox
(left, top, right, bottom) tuple.

blend_mode
Blend mode of this layer. Writable.

Example:

```
from psd_tools.constants import BlendMode
if layer.blend_mode == BlendMode.NORMAL:
    layer.blend_mode = BlendMode.SCREEN
```

Returns *BlendMode*.

bottom
Bottom coordinate.
Returns *int*

clip_layers
Clip layers associated with this layer.

To compose clipping layers:

```
from psd_tools import compose
clip_mask = compose(layer.clip_layers)
```

Returns list of layers

compose (*args, **kwargs)

Compose layer and masks (mask, vector mask, and clipping layers).

Returns PIL. Image, or *None* if the layer has no pixel.

effects

Layer effects.

Returns *Effects*

has_clip_layers ()

Returns True if the layer has associated clipping.

Returns *bool*

has_effects ()

Returns True if the layer has effects.

Returns *bool*

has_mask ()

Returns True if the layer has a mask.

Returns *bool*

has_origination ()

Returns True if the layer has live shape properties.

Returns *bool*

has_pixels ()

Returns True if the layer has associated pixels. When this is True, `topil` method returns PIL. Image.

Returns *bool*

has_vector_mask ()

Returns True if the layer has a vector mask.

Returns *bool*

height

Height of the layer.

Returns *int*

is_group ()

Return True if the layer is a group.

Returns *bool*

is_visible ()

Layer visibility. Takes group visibility in account.

Returns *bool*

kind

Kind of this layer, either of group, pixel, shape, type, smartobject, or psdimage.

Returns *str*

layer_id

Layer ID.

Returns int layer id. if the layer is not assigned an id, -1.

left

Left coordinate. Writable.

Returns int

mask

Returns mask associated with this layer.

Returns *Mask* or *None*

name

Layer name. Writable.

Returns *str*

offset

(left, top) tuple. Writable.

Returns *tuple*

opacity

Opacity of this layer in [0, 255] range. Writable.

Returns int

origination

Property for a list of live shapes or a line.

Some of the vector masks have associated live shape properties, that are Photoshop feature to handle primitive shapes such as a rectangle, an ellipse, or a line. Vector masks without live shape properties are plain path objects.

Returns List of *Invalidated*, *Rectangle*, *RoundedRectangle*, *Ellipse*, or *Line*.

parent

Parent of this layer.

right

Right coordinate.

Returns int

size

(width, height) tuple.

Returns *tuple*

top

Top coordinate. Writable.

Returns int

topil ()

Get PIL Image of the layer.

Returns *PIL*. Image, or *None* if the layer has no pixels.

vector_mask

Returns vector mask associated with this layer.

Returns *VectorMask* or *None*

visible

Layer visibility. Doesn't take group visibility in account. Writable.

Returns *bool*

width

Width of the layer.

Returns *int*

3.4.3 ShapeLayer

class `psd_tools.api.layers.ShapeLayer` (*psd, record, channels, parent*)

Layer that has drawing in vector mask.

bbox

(left, top, right, bottom) tuple.

blend_mode

Blend mode of this layer. Writable.

Example:

```
from psd_tools.constants import BlendMode
if layer.blend_mode == BlendMode.NORMAL:
    layer.blend_mode = BlendMode.SCREEN
```

Returns *BlendMode*.

bottom

Bottom coordinate.

Returns *int*

clip_layers

Clip layers associated with this layer.

To compose clipping layers:

```
from psd_tools import compose
clip_mask = compose(layer.clip_layers)
```

Returns list of layers

compose (**args, **kwargs*)

Compose layer and masks (mask, vector mask, and clipping layers).

Returns *PIL*. Image, or *None* if the layer has no pixel.

effects

Layer effects.

Returns *Effects*

has_clip_layers ()

Returns True if the layer has associated clipping.

Returns *bool*

has_effects ()

Returns True if the layer has effects.

Returns *bool*

has_mask ()

Returns True if the layer has a mask.

Returns *bool*

has_origination ()

Returns True if the layer has live shape properties.

Returns *bool*

has_pixels ()

Returns True if the layer has associated pixels. When this is True, `topil` method returns `PIL.Image`.

Returns *bool*

has_stroke ()

Returns True if the shape has a stroke.

has_vector_mask ()

Returns True if the layer has a vector mask.

Returns *bool*

height

Height of the layer.

Returns *int*

is_group ()

Return True if the layer is a group.

Returns *bool*

is_visible ()

Layer visibility. Takes group visibility in account.

Returns *bool*

kind

Kind of this layer, either of group, pixel, shape, type, smartobject, or psdimage.

Returns *str*

layer_id

Layer ID.

Returns *int* layer id. if the layer is not assigned an id, -1.

left

Left coordinate. Writable.

Returns *int*

mask

Returns mask associated with this layer.

Returns *Mask* or *None*

name

Layer name. Writable.

Returns *str*

offset

(left, top) tuple. Writable.

Returns *tuple*

opacity

Opacity of this layer in [0, 255] range. Writable.

Returns *int*

origination

Property for a list of live shapes or a line.

Some of the vector masks have associated live shape properties, that are Photoshop feature to handle primitive shapes such as a rectangle, an ellipse, or a line. Vector masks without live shape properties are plain path objects.

Returns List of *Invalidated*, *Rectangle*, *RoundedRectangle*, *Ellipse*, or *Line*.

parent

Parent of this layer.

right

Right coordinate.

Returns *int*

size

(width, height) tuple.

Returns *tuple*

stroke

Property for strokes.

top

Top coordinate. Writable.

Returns *int*

topil ()

Get PIL Image of the layer.

Returns *PIL*. Image, or *None* if the layer has no pixels.

vector_mask

Returns vector mask associated with this layer.

Returns *VectorMask* or *None*

visible

Layer visibility. Doesn't take group visibility in account. Writable.

Returns *bool*

width

Width of the layer.

Returns *int*

3.4.4 SmartObjectLayer

class `psd_tools.api.layers.SmartObjectLayer` (*psd, record, channels, parent*)

Layer that inserts external data.

Use `smart_object` attribute to get the external data. See `SmartObject`.

Example:

```
import io
if layer.smart_object.filetype == 'jpg':
    image = Image.open(io.BytesIO(layer.smart_object.data))
```

bbox

(left, top, right, bottom) tuple.

blend_mode

Blend mode of this layer. Writable.

Example:

```
from psd_tools.constants import BlendMode
if layer.blend_mode == BlendMode.NORMAL:
    layer.blend_mode = BlendMode.SCREEN
```

Returns `BlendMode`.

bottom

Bottom coordinate.

Returns `int`

clip_layers

Clip layers associated with this layer.

To compose clipping layers:

```
from psd_tools import compose
clip_mask = compose(layer.clip_layers)
```

Returns list of layers

compose (*args, **kwargs)

Compose layer and masks (mask, vector mask, and clipping layers).

Returns `PIL.Image`, or `None` if the layer has no pixel.

effects

Layer effects.

Returns `Effects`

has_clip_layers ()

Returns True if the layer has associated clipping.

Returns `bool`

has_effects ()

Returns True if the layer has effects.

Returns `bool`

has_mask ()

Returns True if the layer has a mask.

Returns `bool`

has_origination()

Returns True if the layer has live shape properties.

Returns *bool*

has_pixels()

Returns True if the layer has associated pixels. When this is True, `topil` method returns `PIL.Image`.

Returns *bool*

has_vector_mask()

Returns True if the layer has a vector mask.

Returns *bool*

height

Height of the layer.

Returns *int*

is_group()

Return True if the layer is a group.

Returns *bool*

is_visible()

Layer visibility. Takes group visibility in account.

Returns *bool*

kind

Kind of this layer, either of group, pixel, shape, type, smartobject, or psdimage.

Returns *str*

layer_id

Layer ID.

Returns *int* layer id. if the layer is not assigned an id, -1.

left

Left coordinate. Writable.

Returns *int*

mask

Returns mask associated with this layer.

Returns *Mask* or *None*

name

Layer name. Writable.

Returns *str*

offset

(left, top) tuple. Writable.

Returns *tuple*

opacity

Opacity of this layer in [0, 255] range. Writable.

Returns *int*

origination

Property for a list of live shapes or a line.

Some of the vector masks have associated live shape properties, that are Photoshop feature to handle primitive shapes such as a rectangle, an ellipse, or a line. Vector masks without live shape properties are plain path objects.

Returns List of *Invalidated*, *Rectangle*, *RoundedRectangle*, *Ellipse*, or *Line*.

parent

Parent of this layer.

right

Right coordinate.

Returns int

size

(width, height) tuple.

Returns *tuple*

smart_object

Associated smart object.

Returns *SmartObject*.

top

Top coordinate. Writable.

Returns int

topil ()

Get PIL Image of the layer.

Returns *PIL*. Image, or *None* if the layer has no pixels.

vector_mask

Returns vector mask associated with this layer.

Returns *VectorMask* or *None*

visible

Layer visibility. Doesn't take group visibility in account. Writable.

Returns *bool*

width

Width of the layer.

Returns int

3.4.5 TypeLayer

class psd_tools.api.layers.**TypeLayer** (*args)

Layer that has text and styling information for fonts or paragraphs.

Text is accessible at *text* property. Styling information for paragraphs is in *engine_dict*. Document styling information such as font list is in *resource_dict*.

Currently, textual information is read-only.

Example:

```

if layer.kind == 'type':
    print(layer.text)
    print(layer.engine_dict['StyleRun'])

    # Extract font for each substring in the text.
    text = layer.engine_dict['Editor']['Text'].value
    fontset = layer.resource_dict['FontSet']
    runlength = layer.engine_dict['StyleRun']['RunLengthArray']
    rundata = layer.engine_dict['StyleRun']['RunArray']
    index = 0
    for length, style in zip(runlength, rundata):
        substring = text[index:index + length]
        stylesheet = style['StyleSheet']['StyleSheetData']
        font = fontset[stylesheet['Font']]
        print('%r gets %s' % (substring, font))
        index += length

```

bbox

(left, top, right, bottom) tuple.

blend_mode

Blend mode of this layer. Writable.

Example:

```

from psd_tools.constants import BlendMode
if layer.blend_mode == BlendMode.NORMAL:
    layer.blend_mode = BlendMode.SCREEN

```

Returns *BlendMode*.

bottom

Bottom coordinate.

Returns int

clip_layers

Clip layers associated with this layer.

To compose clipping layers:

```

from psd_tools import compose
clip_mask = compose(layer.clip_layers)

```

Returns list of layers

compose (*args, **kwargs)

Compose layer and masks (mask, vector mask, and clipping layers).

Returns PIL. Image, or *None* if the layer has no pixel.

document_resources

Resource set relevant to the document.

effects

Layer effects.

Returns *Effects*

engine_dict

Styling information dict.

has_clip_layers ()

Returns True if the layer has associated clipping.

Returns *bool*

has_effects ()

Returns True if the layer has effects.

Returns *bool*

has_mask ()

Returns True if the layer has a mask.

Returns *bool*

has_origination ()

Returns True if the layer has live shape properties.

Returns *bool*

has_pixels ()

Returns True if the layer has associated pixels. When this is True, `topil` method returns `PIL.Image`.

Returns *bool*

has_vector_mask ()

Returns True if the layer has a vector mask.

Returns *bool*

height

Height of the layer.

Returns *int*

is_group ()

Return True if the layer is a group.

Returns *bool*

is_visible ()

Layer visibility. Takes group visibility in account.

Returns *bool*

kind

Kind of this layer, either of group, pixel, shape, type, smartobject, or psdimage.

Returns *str*

layer_id

Layer ID.

Returns *int* layer id. if the layer is not assigned an id, -1.

left

Left coordinate. Writable.

Returns *int*

mask

Returns mask associated with this layer.

Returns *Mask* or *None*

name

Layer name. Writable.

Returns *str*

offset

(left, top) tuple. Writable.

Returns *tuple*

opacity

Opacity of this layer in [0, 255] range. Writable.

Returns *int*

origination

Property for a list of live shapes or a line.

Some of the vector masks have associated live shape properties, that are Photoshop feature to handle primitive shapes such as a rectangle, an ellipse, or a line. Vector masks without live shape properties are plain path objects.

Returns List of *Invalidated*, *Rectangle*, *RoundedRectangle*, *Ellipse*, or *Line*.

parent

Parent of this layer.

resource_dict

Resource set.

right

Right coordinate.

Returns *int*

size

(width, height) tuple.

Returns *tuple*

text

Text in the layer. Read-only.

Note: New-line character in Photoshop is ‘\r’.

top

Top coordinate. Writable.

Returns *int*

topil ()

Get PIL Image of the layer.

Returns *PIL*. Image, or *None* if the layer has no pixels.

transform

Matrix (xx, xy, yx, yy, tx, ty) applies affine transformation.

vector_mask

Returns vector mask associated with this layer.

Returns *VectorMask* or *None*

visible

Layer visibility. Doesn't take group visibility in account. Writable.

Returns *bool*

warp

Warp configuration.

width

Width of the layer.

Returns *int*

3.5 psd_tools.api.mask

Mask module.

3.5.1 Mask

class `psd_tools.api.mask.Mask` (*layer*)

Mask data attached to a layer.

There are two distinct internal mask data: user mask and vector mask. User mask refers any pixel-based mask whereas vector mask refers a mask from a shape path. Internally, two masks are combined and referred real mask.

background_color

Background color.

bbox

BBox

bottom

Bottom coordinate.

disabled

Disabled.

flags

Flags.

height

Height.

left

Left coordinate.

parameters

Parameters.

real_flags

Real flag.

right

Right coordinate.

size

(Width, Height) tuple.

top

Top coordinate.

topil()

Get PIL Image of the mask.

Returns PIL Image object, or None if the mask is empty.**width**

Width.

3.6 psd_tools.api.shape

Shape module.

3.6.1 VectorMask

class `psd_tools.api.shape.VectorMask` (*data*)

Vector mask data.

Vector mask is a resolution-independent mask that consists of one or more Path objects. In Photoshop, all the path objects are represented as Bezier curves. Check *paths* property for how to deal with path objects.

bbox

Bounding box tuple (left, top, right, bottom) in relative coordinates, where top-left corner is (0., 0.) and bottom-right corner is (1., 1.).

Returns *tuple***clipboard_record**

Clipboard record containing bounding box information.

Depending on the Photoshop version, this field can be *None*.

disabled

If the mask is disabled.

initial_fill_rule

Initial fill rule.

When 0, fill inside of the path. When 1, fill outside of the shape.

Returns *int***inverted**

Invert the mask.

not_linked

If the knots are not linked.

paths

List of *Subpath*. Subpath is a list-like structure that contains one or more *Knot* items. Knot contains relative coordinates of control points for a Bezier curve. *index* indicates which origination item the subpath belongs, and *operation* indicates how to combine multiple shape paths.

In PSD, path fill rule is even-odd.

Example:

```
for subpath in layer.vector_mask.paths:
    anchors = [(
        int(knot.anchor[1] * psd.width),
        int(knot.anchor[0] * psd.height),
    ) for knot in subpath]
```

Returns List of Subpath.

3.6.2 Stroke

class `psd_tools.api.shape.Stroke` (*data*)

Stroke contains decorative information for strokes.

This is a thin wrapper around *Descriptor* structure. Check *_data* attribute to get the raw data.

blend_mode

Blend mode.

content

Fill effect.

enabled

If the stroke is enabled.

fill_enabled

If the stroke fill is enabled.

line_alignment

Alignment, one of *inner*, *outer*, *center*.

line_cap_type

Cap type, one of *butt*, *round*, *square*.

line_dash_offset

Line dash offset in float.

Returns float

line_dash_set

Line dash set in list of `UnitFloat`.

Returns list

line_join_type

Join type, one of *miter*, *round*, *bevel*.

line_width

Stroke width in float.

miter_limit

Miter limit in float.

opacity

Opacity value.

stroke_adjust

Stroke adjust

3.6.3 Origination

Origination keeps live shape properties for some of the primitive shapes. Origination objects are accessible via `origination` property of layers. Following primitive shapes are defined: *Invalidated*, *Line*, *Rectangle*, *Ellipse*, and *RoundedRectangle*.

class `psd_tools.api.shape.Invalidated` (*data*)

Invalidated live shape.

This equals to a primitive shape that does not provide Live shape properties. Use *VectorMask* to access shape information instead of this origination object.

invalidated

Returns *bool*

class `psd_tools.api.shape.Line` (*data*)

Line live shape.

arrow_conc

Returns *int*

arrow_end

Line arrow end.

Returns *bool*

arrow_length

Line arrow length.

Returns *float*

arrow_start

Line arrow start.

Returns *bool*

arrow_width

Line arrow width.

Returns *float*

bbox

Bounding box of the live shape.

Returns *Descriptor*

index

Origination item index.

Returns *int*

invalidated

Returns *bool*

line_end

Line end.

Returns *Descriptor*

line_start

Line start.

Returns *Descriptor*

line_weight
Line weight

Returns *float*

origin_type
Type of the vector shape.

- 1: *Rectangle*
- 2: *RoundedRectangle*
- 4: *Line*
- 5: *Ellipse*

Returns *int*

resolution
Resolution.

Returns *float*

class `psd_tools.api.shape.Ellipse` (*data*)
Ellipse live shape.

bbox
Bounding box of the live shape.

Returns *Descriptor*

index
Origination item index.

Returns *int*

invalidated

Returns *bool*

origin_type
Type of the vector shape.

- 1: *Rectangle*
- 2: *RoundedRectangle*
- 4: *Line*
- 5: *Ellipse*

Returns *int*

resolution
Resolution.

Returns *float*

class `psd_tools.api.shape.Rectangle` (*data*)
Rectangle live shape.

bbox
Bounding box of the live shape.

Returns *Descriptor*

index

Origination item index.

Returns *int*

invalidated

Returns *bool*

origin_type

Type of the vector shape.

- 1: *Rectangle*
- 2: *RoundedRectangle*
- 4: *Line*
- 5: *Ellipse*

Returns *int*

resolution

Resolution.

Returns *float*

class `psd_tools.api.shape.RoundedRectangle` (*data*)

Rounded rectangle live shape.

bbox

Bounding box of the live shape.

Returns *Descriptor*

index

Origination item index.

Returns *int*

invalidated

Returns *bool*

origin_type

Type of the vector shape.

- 1: *Rectangle*
- 2: *RoundedRectangle*
- 4: *Line*
- 5: *Ellipse*

Returns *int*

radii

Corner radii of rounded rectangles. The order is top-left, top-right, bottom-left, bottom-right.

Returns *Descriptor*

resolution

Resolution.

Returns *float*

3.7 psd_tools.api.smart_object

Smart object module.

3.7.1 SmartObject

```
class psd_tools.api.smart_object.SmartObject (layer)
    Smart object that represents embedded or external file.

    data
        Embedded file content, or empty if kind is external or alias

    filename
        Original file name of the object.

    filesize
        File size of the object.

    filetype
        Preferred file extension, such as jpg.

    is_psd()
        Return True if the file is embedded PSD/PSB.

    kind
        Kind of the link, 'data', 'alias', or 'external'.

    open (external_dir=None)
        Open the smart object as binary IO.

        Parameters external_dir – Path to the directory of the external file.

    resolution
        Resolution of the object.

    save (filename=None)
        Save the smart object to a file.

        Parameters filename – File name to export. If None, use the embedded name.

    unique_id
        UUID of the object.

    warp
        Warp parameters.
```

3.8 psd_tools.constants

Various constants for psd_tools

3.8.1 BlendMode

```
class psd_tools.constants.BlendMode
    Blend modes.

    COLOR = b'colr'
```

```

COLOR_BURN = b'idiv'
COLOR_DODGE = b'div '
DARKEN = b'dark'
DARKER_COLOR = b'dkCl'
DIFFERENCE = b'diff'
DISSOLVE = b'diss'
DIVIDE = b'fdiv'
EXCLUSION = b'smud'
HARD_LIGHT = b'hLit'
HARD_MIX = b'hMix'
HUE = b'hue '
LIGHTEN = b'lite'
LIGHTER_COLOR = b'lgCl'
LINEAR_BURN = b'lbrn'
LINEAR_DODGE = b'lddg'
LINEAR_LIGHT = b'lLit'
LUMINOSITY = b'lum '
MULTIPLY = b'mul '
NORMAL = b'norm'
OVERLAY = b'over'
PASS_THROUGH = b'pass'
PIN_LIGHT = b'pLit'
SATURATION = b'sat '
SCREEN = b'scrn'
SOFT_LIGHT = b'sLit'
SUBTRACT = b'fsub'
VIVID_LIGHT = b'vLit'

```

3.8.2 ChannelID

```

class psd_tools.constants.ChannelID
    Channel types.

```

```

CHANNEL_0 = 0
CHANNEL_1 = 1
CHANNEL_2 = 2
CHANNEL_3 = 3
CHANNEL_4 = 4

```

```
CHANNEL_5 = 5
CHANNEL_6 = 6
CHANNEL_7 = 7
CHANNEL_8 = 8
CHANNEL_9 = 9
REAL_USER_LAYER_MASK = -3
TRANSPARENCY_MASK = -1
USER_LAYER_MASK = -2
```

3.8.3 Clipping

```
class psd_tools.constants.Clipping
    Clipping.
    BASE = 0
    NON_BASE = 1
```

3.8.4 ColorMode

```
class psd_tools.constants.ColorMode
    Color mode.
    BITMAP = 0
    CMYK = 4
    DUOTONE = 8
    GRAYSCALE = 1
    INDEXED = 2
    LAB = 9
    MULTICHANNEL = 7
    RGB = 3
```

3.8.5 ColorSpaceID

```
class psd_tools.constants.ColorSpaceID
    Color space types.
    CMYK = 2
    GRAYSCALE = 8
    HSB = 1
    LAB = 7
    RGB = 0
```

3.8.6 Compression

class `psd_tools.constants.Compression`

Compression modes.

Compression. 0 = Raw Data, 1 = RLE compressed, 2 = ZIP without prediction, 3 = ZIP with prediction.

PACK_BITS = 1

RAW = 0

ZIP = 2

ZIP_WITH_PREDICTION = 3

3.8.7 EffectOSType

class `psd_tools.constants.EffectOSType`

OS Type keys for Layer Effects.

BEVEL = `b'bevl'`

COMMON_STATE = `b'cmnS'`

DROP_SHADOW = `b'dsdw'`

INNER_GLOW = `b'iglw'`

INNER_SHADOW = `b'isdw'`

OUTER_GLOW = `b'oglw'`

SOLID_FILL = `b'sofi'`

3.8.8 GlobalLayerMaskKind

class `psd_tools.constants.GlobalLayerMaskKind`

Global layer mask kind.

COLOR_PROTECTED = 1

COLOR_SELECTED = 0

PER_LAYER = 128

3.8.9 LinkedLayerType

class `psd_tools.constants.LinkedLayerType`

Linked layer types.

ALIAS = `b'liFA'`

DATA = `b'liFD'`

EXTERNAL = `b'liFE'`

3.8.10 ImageResourceID

class `psd_tools.constants.ImageResourceID`

Image resource keys.

Note the following is not defined for performance reasons.

- `PATH_INFO_10` to `PATH_INFO_989` corresponding to 2010 - 2989
- **`PLUGIN_RESOURCES_10` to `PLUGIN_RESOURCES_989` corresponding to 4010 - 4989**

```
ALPHA_IDENTIFIERS = 1053
ALPHA_NAMES_PASCAL = 1006
ALPHA_NAMES_UNICODE = 1045
ALTERNATE_DUOTONE_COLORS = 1066
ALTERNATE_SPOT_COLORS = 1067
AUTO_SAVE_FILE_PATH = 1086
AUTO_SAVE_FORMAT = 1087
BACKGROUND_COLOR = 1010
BORDER_INFO = 1009
CAPTION_DIGEST = 1061
CAPTION_PASCAL = 1008
CLIPPING_PATH_NAME = 2999
COLOR_HALFTONING_INFO = 1013
COLOR_SAMPLERS_RESOURCE = 1073
COLOR_SAMPLERS_RESOURCE_OBSOLETE = 1038
COLOR_TRANSFER_FUNCTION = 1016
COPYRIGHT_FLAG = 1034
COUNT_INFO = 1080
DISPLAY_INFO = 1077
DISPLAY_INFO_OBSOLETE = 1007
DUOTONE_HALFTONING_INFO = 1014
DUOTONE_IMAGE_INFO = 1018
DUOTONE_TRANSFER_FUNCTION = 1017
EFFECTIVE_BW = 1019
EFFECTS_VISIBLE = 1042
EPS_OPTIONS = 1021
EXIF_DATA_1 = 1058
EXIF_DATA_3 = 1059
GLOBAL_ALTITUDE = 1049
GLOBAL_ANGLE = 1037
```



```
GRAYSCALE_HALFTONING_INFO = 1012
GRAYSCALE_TRANSFER_FUNCTION = 1015
GRID_AND_GUIDES_INFO = 1032
HDR_TONING_INFO = 1070
ICC_PROFILE = 1039
ICC_UNTAGGED_PROFILE = 1041
IDS_SEED_NUMBER = 1044
IMAGE_MODE_RAW = 1029
IMAGE_READY_DATA_SETS = 7001
IMAGE_READY_VARIABLES = 7000
INDEXED_COLOR_TABLE_COUNT = 1046
IPTC_NAA = 1028
JPEG_QUALITY = 1030
JUMP_TO_XPEP = 1052
LAYER_COMPS = 1065
LAYER_GROUPS_ENABLED_ID = 1072
LAYER_GROUP_INFO = 1026
LAYER_SELECTION_IDS = 1069
LAYER_STATE_INFO = 1024
LIGHTROOM_WORKFLOW = 8000
MAC_NSPRINTINFO = 1084
MAC_PRINT_MANAGER_INFO = 1001
MEASUREMENT_SCALE = 1074
OBSOLETE1 = 1000
OBSOLETE2 = 1003
OBSOLETE3 = 1020
OBSOLETE4 = 1023
OBSOLETE5 = 1027
ONION_SKINS = 1078
ORIGIN_PATH_INFO = 3000
PATH_INFO_0 = 2000
PATH_INFO_1 = 2001
PATH_INFO_2 = 2002
PATH_INFO_3 = 2003
PATH_INFO_4 = 2004
PATH_INFO_5 = 2005
```

PATH_INFO_6 = 2006
PATH_INFO_7 = 2007
PATH_INFO_8 = 2008
PATH_INFO_9 = 2009
PATH_INFO_990 = 2990
PATH_INFO_991 = 2991
PATH_INFO_992 = 2992
PATH_INFO_993 = 2993
PATH_INFO_994 = 2994
PATH_INFO_995 = 2995
PATH_INFO_996 = 2996
PATH_INFO_997 = 2997
PATH_SELECTION_STATE = 1088
PIXEL_ASPECT_RATIO = 1064
PLUGIN_RESOURCE_0 = 4000
PLUGIN_RESOURCE_1 = 4001
PLUGIN_RESOURCE_2 = 4002
PLUGIN_RESOURCE_3 = 4003
PLUGIN_RESOURCE_4 = 4004
PLUGIN_RESOURCE_4990 = 4990
PLUGIN_RESOURCE_4991 = 4991
PLUGIN_RESOURCE_4992 = 4992
PLUGIN_RESOURCE_4993 = 4993
PLUGIN_RESOURCE_4994 = 4994
PLUGIN_RESOURCE_4995 = 4995
PLUGIN_RESOURCE_4996 = 4996
PLUGIN_RESOURCE_4997 = 4997
PLUGIN_RESOURCE_4998 = 4998
PLUGIN_RESOURCE_4999 = 4990
PLUGIN_RESOURCE_5 = 4005
PLUGIN_RESOURCE_6 = 4006
PLUGIN_RESOURCE_7 = 4007
PLUGIN_RESOURCE_8 = 4008
PLUGIN_RESOURCE_9 = 4009
PRINT_FLAGS = 1011
PRINT_FLAGS_INFO = 10000

```
PRINT_INFO_CS2 = 1071
PRINT_INFO_CS5 = 1082
PRINT_SCALE = 1062
PRINT_STYLE = 1083
QUICK_MASK_INFO = 1022
RESOLUTION_INFO = 1005
SHEET_DISCLOSURE = 1076
SLICES = 1050
SPOT_HALFTONE = 1043
THUMBNAIL_RESOURCE = 1036
THUMBNAIL_RESOURCE_PS4 = 1033
TIMELINE_INFO = 1075
TRANSPARENCY_INDEX = 1047
URL = 1035
URL_LIST = 1054
VERSION_INFO = 1057
WATERMARK = 1040
WINDOWS_DEVMODE = 1085
WORKFLOW_URL = 1051
WORKING_PATH = 1025
XMP_METADATA = 1060
```

3.8.11 PathResourceID

class `psd_tools.constants.PathResourceID`

An enumeration.

```
CLIPBOARD = 7
CLOSED_KNOT_LINKED = 1
CLOSED_KNOT_UNLINKED = 2
CLOSED_LENGTH = 0
INITIAL_FILL = 8
OPEN_KNOT_LINKED = 4
OPEN_KNOT_UNLINKED = 5
OPEN_LENGTH = 3
PATH_FILL = 6
```

3.8.12 PlacedLayerType

```
class psd_tools.constants.PlacedLayerType
    An enumeration.

    IMAGE_STACK = 3

    RASTER = 2

    UNKNOWN = 0

    VECTOR = 1
```

3.8.13 PrintScaleStyle

```
class psd_tools.constants.PrintScaleStyle
    Print scale style.

    CENTERED = 0

    SIZE_TO_FIT = 1

    USER_DEFINED = 2
```

3.8.14 SectionDivider

```
class psd_tools.constants.SectionDivider
    An enumeration.

    BOUNDING_SECTION_DIVIDER = 3

    CLOSED_FOLDER = 2

    OPEN_FOLDER = 1

    OTHER = 0
```

3.8.15 TaggedBlockID

```
class psd_tools.constants.TaggedBlockID
    Tagged blocks keys.

    ALPHA = b'Alph'

    ANIMATION_EFFECTS = b'anFX'

    ANNOTATIONS = b'Anno'

    ARTBOARD_DATA1 = b'artb'

    ARTBOARD_DATA2 = b'artd'

    ARTBOARD_DATA3 = b'abdd'

    BLACK_AND_WHITE = b'blwh'

    BLEND_CLIPPING_ELEMENTS = b'clbl'

    BLEND_FILL_OPACITY = b'iOpa'

    BLEND_INTERIOR_ELEMENTS = b'infx'
```

```
BRIGHTNESS_AND_CONTRAST = b'brit'  
CHANNEL_BLENDING_RESTRICTIONS_SETTING = b'brst'  
CHANNEL_MIXER = b'mixr'  
COLOR_BALANCE = b'blnc'  
COLOR_LOOKUP = b'clrL'  
COMPUTER_INFO = b'cinf'  
CONTENT_GENERATOR_EXTRA_DATA = b'CgEd'  
CURVES = b'curv'  
EFFECTS_LAYER = b'lrFX'  
EXPORT_SETTING1 = b'extd'  
EXPORT_SETTING2 = b'extn'  
EXPOSURE = b'expA'  
FILTER_EFFECTS1 = b'FXid'  
FILTER_EFFECTS2 = b'FEid'  
FILTER_EFFECTS3 = b'FELS'  
FILTER_MASK = b'FMsk'  
FOREIGN_EFFECT_ID = b'ffxi'  
GRADIENT_FILL_SETTING = b'GdFl'  
GRADIENT_MAP = b'grdm'  
HUE_SATURATION = b'hue2'  
HUE_SATURATION_V4 = b'hue '  
INVERT = b'nvrt'  
KNOCKOUT_SETTING = b'knko'  
LAYER = b'Layr'  
LAYER_16 = b'Lr16'  
LAYER_32 = b'Lr32'  
LAYER_ID = b'lyid'  
LAYER_MASK_AS_GLOBAL_MASK = b'lmgm'  
LAYER_NAME_SOURCE_SETTING = b'lnsr'  
LAYER_VERSION = b'lyvr'  
LEVELS = b'levl'  
LINKED_LAYER1 = b'lnkD'  
LINKED_LAYER2 = b'lnk2'  
LINKED_LAYER3 = b'lnk3'  
LINKED_LAYER_EXTERNAL = b'lnkE'  
METADATA_SETTING = b'shmd'
```

```
NESTED_SECTION_DIVIDER_SETTING = b'lSDK'  
OBJECT_BASED_EFFECTS_LAYER_INFO = b'lfx2'  
OBJECT_BASED_EFFECTS_LAYER_INFO_V0 = b'lMfx'  
OBJECT_BASED_EFFECTS_LAYER_INFO_V1 = b'lfxs'  
PATTERNS1 = b'Patt'  
PATTERNS2 = b'Pat2'  
PATTERNS3 = b'Pat3'  
PATTERN_DATA = b'shpa'  
PATTERN_FILL_SETTING = b'PtF1'  
PHOTO_FILTER = b'phf1'  
PIXEL_SOURCE_DATA1 = b'PxSc'  
PIXEL_SOURCE_DATA2 = b'PxSD'  
PLACED_LAYER1 = b'plLd'  
PLACED_LAYER2 = b'PlLd'  
POSTERIZE = b'post'  
PROTECTED_SETTING = b'lspf'  
REFERENCE_POINT = b'fxrp'  
SAVING_MERGED_TRANSPARENCY = b'Mtrn'  
SAVING_MERGED_TRANSPARENCY16 = b'Mt16'  
SAVING_MERGED_TRANSPARENCY32 = b'Mt32'  
SECTION_DIVIDER_SETTING = b'lSct'  
SELECTIVE_COLOR = b'selc'  
SHEET_COLOR_SETTING = b'lclr'  
SMART_OBJECT_LAYER_DATA1 = b'SoLd'  
SMART_OBJECT_LAYER_DATA2 = b'SoLE'  
SOLID_COLOR_SHEET_SETTING = b'SoCo'  
TEXT_ENGINE_DATA = b'Txt2'  
THRESHOLD = b'thrs'  
TRANSPARENCY_SHAPES_LAYER = b'tsly'  
TYPE_TOOL_INFO = b'tySh'  
TYPE_TOOL_OBJECT_SETTING = b'TySh'  
UNICODE_LAYER_NAME = b'luni'  
UNICODE_PATH_NAME = b'pths'  
USER_MASK = b'lMsk'  
USING_ALIGNED_RENDERING = b'sn2P'  
VECTOR_MASK_AS_GLOBAL_MASK = b'vmgm'
```

```

VECTOR_MASK_SETTING1 = b'vmsk'
VECTOR_MASK_SETTING2 = b'vsms'
VECTOR_ORIGNATION_DATA = b'vogk'
VECTOR_STROKE_CONTENT_DATA = b'vscg'
VECTOR_STROKE_DATA = b'vstk'
VIBRANCE = b'vibA'

```

3.9 psd_tools.psd

Low-level API that translates binary data to Python structure

3.9.1 PSD

```

class psd_tools.psd.PSD(header=NOTHING, color_mode_data=NOTHING, image_resources=NOTHING, layer_and_mask_information=NOTHING, image_data=NOTHING)

```

Low-level PSD file structure that resembles the [specification](#).

Example:

```

from psd_tools.psd import PSD

with open(input_file, 'rb') as f:
    psd = PSD.read(f)

with open(output_file, 'wb') as f:
    psd.write(f)

```

header

See [FileHeader](#).

color_mode_data

See [ColorModeData](#).

image_resources

See [ImageResources](#).

layer_and_mask_information

See [LayerAndMaskInformation](#).

image_data

See [ImageData](#).

color_mode_data

classmethod frombytes (*data*, *args, **kwargs)

Read the element from bytes.

Parameters *data* – bytes

header

image_data

image_resources

layer_and_mask_information

classmethod `read(fp, encoding='macroman', **kwargs)`
Read the element from a file-like object.

Parameters `fp` – file-like object

Return type *PSD*

tobytes (**args*, ***kwargs*)
Write the element to bytes.

Return type bytes

validate ()
Validate the attribute.

write (*fp*, *encoding='macroman'*, ***kwargs*)
Write the element to a file-like object.

3.10 psd_tools.psd.color_mode_data

Color mode data structure.

3.10.1 ColorModeData

class `psd_tools.psd.color_mode_data.ColorModeData (value: bytes = b'')`
Color mode data section of the PSD file.

For indexed color images the data is the color table for the image in a non-interleaved order.

Duotone images also have this data, but the data format is undocumented.

interleave ()

classmethod `read(fp)`
Read the element from a file-like object.

Parameters `fp` – file-like object

Return type *ColorModeData*

value

write (*fp*)
Write the element to a file-like object.

Parameters `fp` – file-like object

3.11 psd_tools.psd.descriptor

Descriptor data structure.

Descriptors are basic data structure used throughout PSD files.

3.11.1 Alias

```
class psd_tools.psd.descriptor.Alias (value: bytes = b'x00x00x00x00')
    Alias structure equivalent to RawData.

    value

    ostype = b'alis'
```

3.11.2 Bool

```
class psd_tools.psd.descriptor.Bool (value=False)
    Bool structure.

    value

    ostype = b'bool'

    classmethod read (fp)
        Read the element from a file-like object.

        Parameters fp – file-like object

    write (fp)
        Write the element to a file-like object.

        Parameters fp – file-like object
```

3.11.3 Class

```
class psd_tools.psd.descriptor.Class (name: str = "", classID: bytes = b'x00x00x00x00')
    Class structure.

    name

    classID

    classmethod read (fp)
        Read the element from a file-like object.

        Parameters fp – file-like object

    write (fp)
        Write the element to a file-like object.

        Parameters fp – file-like object
```

3.11.4 Class1

```
class psd_tools.psd.descriptor.Class1 (name: str = "", classID: bytes = b'x00x00x00x00')
    Class structure equivalent to Class.

    ostype = b'type'
```

3.11.5 Class2

```
class psd_tools.psd.descriptor.Class2 (name: str = ", classID: bytes = b'x00x00x00x00')  
    Class structure equivalent to Class.  
  
    ostype = b'GlbC'
```

3.11.6 Class3

```
class psd_tools.psd.descriptor.Class3 (name: str = ", classID: bytes = b'x00x00x00x00')  
    Class structure equivalent to Class.  
  
    ostype = b'Clss'
```

3.11.7 Descriptor

```
class psd_tools.psd.descriptor.Descriptor (items=NOTHING, name: str = ", classID=<DescriptorClassID.NULL: b'null'>)
```

Dict-like descriptor structure.

Example:

```
for key in descriptor:  
    print(descriptor[key])
```

name

classID

ostype = **b'Objc'**

classmethod read (*fp*)

Read the element from a file-like object.

Parameters **fp** – file-like object

write (*fp*)

Write the element to a file-like object.

Parameters **fp** – file-like object

3.11.8 Double

```
class psd_tools.psd.descriptor.Double (value=0.0)  
    Double structure.
```

value

ostype = **b'doub'**

classmethod read (*fp*)

Read the element from a file-like object.

Parameters **fp** – file-like object

write (*fp*)

Write the element to a file-like object.

Parameters **fp** – file-like object

3.11.9 Enum

```
class psd_tools.psd.descriptor.Enum(TypeID: bytes = b'x00x00x00x00', enum: bytes = b'x00x00x00x00')
    Enum structure.

    value

    ostype = b'enum'

    classmethod read(fp)
        Read the element from a file-like object.

        Parameters fp – file-like object

    write(fp)
        Write the element to a file-like object.

        Parameters fp – file-like object
```

3.11.10 EnumeratedReference

```
class psd_tools.psd.descriptor.EnumeratedReference(name: str = "", classID: bytes = b'x00x00x00x00', typeID: bytes = b'x00x00x00x00', enum: bytes = b'x00x00x00x00')

    Enumerated reference structure.

    value

    ostype = b'Enmr'

    classmethod read(fp)
        Read the element from a file-like object.

        Parameters fp – file-like object

    write(fp)
        Write the element to a file-like object.

        Parameters fp – file-like object
```

3.11.11 GlobalObject

```
class psd_tools.psd.descriptor.GlobalObject(items=NOTHING, name: str = "", classID=<DescriptorClassID.NULL: b'null'>)

    Global object structure equivalent to Descriptor.

    ostype = b'Glb0'
```

3.11.12 Identifier

```
class psd_tools.psd.descriptor.Identifier(value=0)

    Identifier equivalent to Integer.

    ostype = b'Idnt'
```

3.11.13 Index

```
class psd_tools.psd.descriptor.Index (value=0)
    Index equivalent to Integer.

    ostype = b'indx'
```

3.11.14 Integer

```
class psd_tools.psd.descriptor.Integer (value=0)
    Integer structure.

    value

    ostype = b'long'

    classmethod read (fp)
        Read the element from a file-like object.

        Parameters fp – file-like object

    write (fp)
        Write the element to a file-like object.

        Parameters fp – file-like object
```

3.11.15 LargeInteger

```
class psd_tools.psd.descriptor.LargeInteger (value=0)
    LargeInteger structure.

    value

    ostype = b'comp'

    classmethod read (fp)
        Read the element from a file-like object.

        Parameters fp – file-like object

    write (fp)
        Write the element to a file-like object.

        Parameters fp – file-like object
```

3.11.16 List

```
class psd_tools.psd.descriptor.List (items=NOTHING)
    List structure.

    items

    ostype = b'VLLs'

    classmethod read (fp)
        Read the element from a file-like object.

        Parameters fp – file-like object
```

write (*fp*)
Write the element to a file-like object.
Parameters *fp* – file-like object

3.11.17 Name

class `psd_tools.psd.descriptor.Name` (*name: str = "", classID: bytes = b'x00x00x00x00', value: str = ""*)

Name structure (Undocumented).

ostype = `b'name'`

classmethod **read** (*fp*)
Read the element from a file-like object.

Parameters *fp* – file-like object

write (*fp*)
Write the element to a file-like object.

Parameters *fp* – file-like object

3.11.18 ObjectArray

class `psd_tools.psd.descriptor.ObjectArray` (*items=NOTHING, items_count: int = 0, name: str = "", classID=<DescriptorClassID.NULL: b'null'>*)

Object array structure almost equivalent to *Descriptor*.

items_count

name

classID

ostype = `b'ObAr'`

classmethod **read** (*fp*)
Read the element from a file-like object.

Parameters *fp* – file-like object

write (*fp*)
Write the element to a file-like object.

Parameters *fp* – file-like object

3.11.19 Property

class `psd_tools.psd.descriptor.Property` (*name: str = "", classID: bytes = b'x00x00x00x00', keyID: bytes = b'x00x00x00x00'*)

Property structure.

name

ostype = `b'prop'`

classmethod **read** (*fp*)
Read the element from a file-like object.

Parameters **fp** – file-like object

write (*fp*)

Write the element to a file-like object.

Parameters **fp** – file-like object

3.11.20 Offset

class `psd_tools.psd.descriptor.Offset` (*name: str = "*, *classID: bytes = b'x00x00x00x00'*,
value=0)

Offset structure.

value

ostype = `b'rele'`

classmethod **read** (*fp*)

Read the element from a file-like object.

Parameters **fp** – file-like object

write (*fp*)

Write the element to a file-like object.

Parameters **fp** – file-like object

3.11.21 Path

class `psd_tools.psd.descriptor.Path` (*value: bytes = b'x00x00x00x00'*)

Undocumented path structure equivalent to `RawData`.

ostype = `b'Pth '`

3.11.22 RawData

class `psd_tools.psd.descriptor.RawData` (*value: bytes = b'x00x00x00x00'*)

RawData structure.

value

bytes

ostype = `b'tdta'`

classmethod **read** (*fp*)

Read the element from a file-like object.

Parameters **fp** – file-like object

write (*fp*)

Write the element to a file-like object.

Parameters **fp** – file-like object

3.11.23 Reference

```
class psd_tools.psd.descriptor.Reference (items=NOTHING)
    Reference structure equivalent to List.

    ostype = b'obj'
```

3.11.24 String

```
class psd_tools.psd.descriptor.String (value: str = "")
    String structure.

    value

    ostype = b'TEXT'
```

3.11.25 UnitFloat

```
class psd_tools.psd.descriptor.UnitFloat (value: float = 0.0, unit=<UnitFloatType.NONE: b'#Nne'>)
    Unit float structure.

    unit

    value

    ostype = b'UnfF'

    classmethod read (fp)
        Read the element from a file-like object.

        Parameters fp – file-like object

    unit

    value

    write (fp)
        Write the element to a file-like object.

        Parameters fp – file-like object
```

3.11.26 UnitFloats

```
class psd_tools.psd.descriptor.UnitFloats (unit=<UnitFloatType.NONE: b'#Nne'>, values=NOTHING)
    Unit floats structure.

    unit

    values

    ostype = b'Unf1'

    classmethod read (fp)
        Read the element from a file-like object.

        Parameters fp – file-like object

    write (fp)
        Write the element to a file-like object.
```

Parameters `fp` – file-like object

3.12 psd_tools.psd.engine_data

EngineData parser.

PSD file embeds text formatting data in its own markup language referred EngineData. The format looks like the following:

```
<<
/EngineDict
<<
  /Editor
  <<
    /Text (~Make a change and save.)
  >>
>>
/Font
<<
  /Name (~HelveticaNeue-Light)
  /FillColor
  <<
    /Type 1
    /Values [ 1.0 0.0 0.0 0.0 ]
  >>
  /StyleSheetSet [
  <<
    /Name (~Normal RGB)
  >>
  ]
>>
>>
```

3.12.1 EngineData

class `psd_tools.psd.engine_data.EngineData` (*items=NOTHING*)
Dict-like element.

`TYPE_TOOL_OBJECT_SETTING` tagged block contains this object in its descriptor.

3.12.2 EngineData2

class `psd_tools.psd.engine_data.EngineData2` (*items=NOTHING*)
Dict-like element.

`TEXT_ENGINE_DATA` tagged block has this object.

write (*fp, indent=None, write_container=False, **kwargs*)
Write the element to a file-like object.

3.12.3 Bool

class `psd_tools.psd.engine_data.Bool` (*value=False*)
Bool element.

classmethod frombytes (*data*)

Read the element from bytes.

Parameters *data* – bytes

classmethod read (*fp*)

Read the element from a file-like object.

Parameters *fp* – file-like object

write (*fp*, *indent=0*)

Write the element to a file-like object.

3.12.4 Dict

class `psd_tools.psd.engine_data.Dict` (*items=NOTHING*)

Dict-like element.

classmethod frombytes (*data*, ***kwargs*)

Read the element from bytes.

Parameters *data* – bytes

classmethod read (*fp*, ***kwargs*)

Read the element from a file-like object.

Parameters *fp* – file-like object

write (*fp*, *indent=0*, *write_container=True*)

Write the element to a file-like object.

3.12.5 Float

class `psd_tools.psd.engine_data.Float` (*value=0.0*)

Float element.

classmethod frombytes (*data*)

Read the element from bytes.

Parameters *data* – bytes

classmethod read (*fp*)

Read the element from a file-like object.

Parameters *fp* – file-like object

write (*fp*)

Write the element to a file-like object.

3.12.6 Integer

class `psd_tools.psd.engine_data.Integer` (*value=0*)

Integer element.

classmethod frombytes (*data*)

Read the element from bytes.

Parameters *data* – bytes

classmethod read (*fp*)
Read the element from a file-like object.
Parameters **fp** – file-like object

write (*fp, indent=0*)
Write the element to a file-like object.

3.12.7 List

class `psd_tools.psd.engine_data.List` (*items=NOTHING*)
List-like element.

classmethod frombytes (*data*)
Read the element from bytes.
Parameters **data** – bytes

classmethod read (*fp*)
Read the element from a file-like object.
Parameters **fp** – file-like object

write (*fp, indent=None*)
Write the element to a file-like object.

3.12.8 Property

class `psd_tools.psd.engine_data.Property` (*value=None*)
Property element.

classmethod frombytes (*data*)
Read the element from bytes.
Parameters **data** – bytes

classmethod read (*fp*)
Read the element from a file-like object.
Parameters **fp** – file-like object

write (*fp*)
Write the element to a file-like object.

3.12.9 String

class `psd_tools.psd.engine_data.String` (*value=None*)
String element.

classmethod frombytes (*data*)
Read the element from bytes.
Parameters **data** – bytes

classmethod read (*fp*)
Read the element from a file-like object.
Parameters **fp** – file-like object

write (*fp*)
Write the element to a file-like object.

3.13 psd_tools.psd.header

File header structure.

3.13.1 FileHeader

class `psd_tools.psd.header.FileHeader` (*signature: bytes = b'8BPS', version: int = 1, channels: int = 4, height: int = 64, width: int = 64, depth: int = 8, color_mode=<ColorMode.RGB: 3>*)

Header section of the PSD file.

Example:

```
from psd_tools.psd.header import FileHeader
from psd_tools.constants import ColorMode

header = FileHeader(channels=2, height=359, width=400, depth=8,
                    color_mode=ColorMode.GRAYSCALE)
```

signature

Signature: always equal to `b'8BPS'`.

version

Version number. PSD is 1, and PSB is 2.

channels

The number of channels in the image, including any alpha channels.

height

The height of the image in pixels.

width

The width of the image in pixels.

depth

The number of bits per channel.

color_mode

The color mode of the file. See [ColorMode](#)

channels

color_mode

depth

height

classmethod read (*fp*)

Read the element from a file-like object.

Parameters `fp` – file-like object

Return type `FileHeader`

signature

version

width

write (*fp*)

Write the element to a file-like object.

Parameters *fp* – file-like object

3.14 psd_tools.psd.image_data

Image data section structure.

3.14.1 ImageData

class `psd_tools.psd.image_data.ImageData` (*compression=<Compression.RAW: 0>*, *data: bytes = b''*)

Merged channel image data.

compression

See *Compression*.

data

compression

data

get_data (*header*)

Get decompressed data.

Parameters *header* – See *FileHeader*.

Returns list of bytes corresponding each channel.

Return type list

classmethod `new` (*header*, *color=0*, *compression=<Compression.RAW: 0>*)

Create a new image data object.

Parameters

- **header** – FileHeader.
- **compression** – compression type.
- **color** – default color. int or iterable for channel length.

classmethod `read` (*fp*)

Read the element from a file-like object.

Parameters *fp* – file-like object

Return type *ImageData*

set_data (*data*, *header*)

Set raw data and compress.

Parameters

- **data** – list of raw data bytes corresponding channels.
- **compression** – compression type, see *Compression*.

- **header** – See *FileHeader*.

Returns length of compressed data.

write (*fp*)

Write the element to a file-like object.

Parameters *fp* – file-like object

Return type int

3.15 psd_tools.psd.image_resources

Image resources section structure. Image resources are used to store non-pixel data associated with images, such as pen tool paths.

The following resources are plain bytes:

```
ImageResourceID.OBSOLETE1: 1000
ImageResourceID.MAC_PRINT_MANAGER_INFO: 1001
ImageResourceID.OBSOLETE2: 1003
ImageResourceID.DISPLAY_INFO_OBSOLETE: 1007
ImageResourceID.BORDER_INFO: 1009
ImageResourceID.DUOTONE_IMAGE_INFO: 1018
ImageResourceID.EFFECTIVE_BW: 1019
ImageResourceID.OBSOLETE3: 1020
ImageResourceID.EPS_OPTIONS: 1021
ImageResourceID.QUICK_MASK_INFO: 1022
ImageResourceID.OBSOLETE4: 1023
ImageResourceID.WORKING_PATH: 1025
ImageResourceID.OBSOLETE5: 1027
ImageResourceID.IPTC_NAA: 1028
ImageResourceID.IMAGE_MODE_RAW: 1029
ImageResourceID.JPEG_QUALITY: 1030
ImageResourceID.URL: 1035
ImageResourceID.COLOR_SAMPLERS_RESOURCE_OBSOLETE: 1038
ImageResourceID.ICC_PROFILE: 1039
ImageResourceID.SPOT_HALFTONE: 1043
ImageResourceID.JUMP_TO_XPEP: 1052
ImageResourceID.EXIF_DATA_1: 1058
ImageResourceID.EXIF_DATA_3: 1059
ImageResourceID.XMP_METADATA: 1060
ImageResourceID.CAPTION_DIGEST: 1061
ImageResourceID.ALTERNATE_DUOTONE_COLORS: 1066
ImageResourceID.ALTERNATE_SPOT_COLORS: 1067
ImageResourceID.HDR_TONING_INFO: 1070
ImageResourceID.PRINT_INFO_CS2: 1071
ImageResourceID.COLOR_SAMPLERS_RESOURCE: 1073
ImageResourceID.DISPLAY_INFO: 1077
ImageResourceID.MAC_NSPRINTINFO: 1084
ImageResourceID.WINDOWS_DEVMODE: 1085
ImageResourceID.PATH_INFO_N: 2000-2999
ImageResourceID.PLUGIN_RESOURCES_N: 4000-4999
ImageResourceID.IMAGE_READY_VARIABLES: 7000
ImageResourceID.IMAGE_READY_DATA_SETS: 7001
ImageResourceID.LIGHTROOM_WORKFLOW: 8000
```

3.15.1 ImageResources

class `psd_tools.psd.image_resources.ImageResources` (*items=NOTHING*)
Image resources section of the PSD file. Dict of *ImageResource*.

enum
alias of `psd_tools.constants.ImageResourceID`

get_data (*key, default=None*)
Get data from the image resources.

Shortcut for the following:

```
if key in image_resources:  
    value = tagged_blocks[key].data
```

classmethod new (***kwargs*)
Create a new default image resources.

Returns *ImageResources*

classmethod read (*fp, encoding='macroman'*)
Read the element from a file-like object.

Parameters *fp* – file-like object

Return type *ImageResources*

write (*fp, encoding='macroman'*)
Write the element to a file-like object.

Parameters *fp* – file-like object

3.15.2 ImageResource

class `psd_tools.psd.image_resources.ImageResource` (*signature: bytes = b'8BIM', key: int = 1000, name: str = "", data: bytes = b""*)

Image resource block.

signature
Binary signature, always `b'8BIM'`.

key
Unique identifier for the resource. See *ImageResourceID*.

name

data
The resource data.

data

key

name

classmethod read (*fp, encoding='macroman'*)
Read the element from a file-like object.

Parameters *fp* – file-like object

Return type *ImageResource*

signature

write (*fp*, *encoding='macroman'*)

Write the element to a file-like object.

Parameters **fp** – file-like object

Return type int

3.16 psd_tools.psd.layer_and_mask

Layer and mask data structure.

3.16.1 LayerAndMaskInformation

class `psd_tools.psd.layer_and_mask.LayerAndMaskInformation` (*layer_info=None*,
global_layer_mask_info=None,
tagged_blocks=None)

Layer and mask information section.

layer_info

See *LayerInfo*.

global_layer_mask_info

See *GlobalLayerMaskInfo*.

tagged_blocks

See *TaggedBlocks*.

global_layer_mask_info

layer_info

classmethod **read** (*fp*, *encoding='macroman'*, *version=1*)

Read the element from a file-like object.

Parameters

- **fp** – file-like object
- **encoding** – encoding of the string
- **version** – psd file version

Return type *LayerAndMaskInformation*

tagged_blocks

write (*fp*, *encoding='macroman'*, *version=1*, *padding=4*)

Write the element to a file-like object.

Parameters

- **fp** – file-like object
- **encoding** – encoding of the string
- **version** – psd file version

3.16.2 LayerInfo

class `psd_tools.psd.layer_and_mask.LayerInfo` (*layer_count: int = 0, layer_records=None, channel_image_data=None*)

High-level organization of the layer information.

layer_count

Layer count. If it is a negative number, its absolute value is the number of layers and the first alpha channel contains the transparency data for the merged result.

layer_records

Information about each layer. See *LayerRecords*.

channel_image_data

Channel image data. Contains one or more image data records. See *ChannelImageData*.

channel_image_data

layer_count

layer_records

classmethod `read` (*fp, encoding='macroman', version=1*)

Read the element from a file-like object.

Parameters

- **fp** – file-like object
- **encoding** – encoding of the string
- **version** – psd file version

Return type *LayerInfo*

write (*fp, encoding='macroman', version=1, padding=4*)

Write the element to a file-like object.

Parameters **fp** – file-like object

Return type `int`

3.16.3 GlobalLayerMaskInfo

class `psd_tools.psd.layer_and_mask.GlobalLayerMaskInfo` (*overlay_color=None, opacity: int = 0, kind=<GlobalLayerMaskKind.PER_LAYER: 128>*)

Global mask information.

overlay_color

Overlay color space (undocumented) and color components.

opacity

Opacity. 0 = transparent, 100 = opaque.

kind

Kind. 0 = Color selected–i.e. inverted; 1 = Color protected; 128 = use value stored per layer. This value is preferred. The others are for backward compatibility with beta versions.

kind

opacity

overlay_color

classmethod read (*fp*)

Read the element from a file-like object.

Parameters *fp* – file-like object

Return type *GlobalLayerMaskInfo*

write (*fp*)

Write the element to a file-like object.

Parameters *fp* – file-like object

3.16.4 LayerRecords

class `psd_tools.psd.layer_and_mask.LayerRecords` (*items=NOTHING*)

List of layer records. See *LayerRecord*.

classmethod read (*fp, layer_count, encoding='macroman', version=1*)

Read the element from a file-like object.

Parameters *fp* – file-like object

Return type *LayerRecords*

3.16.5 LayerRecord

class `psd_tools.psd.layer_and_mask.LayerRecord` (*top: int = 0, left: int = 0, bottom: int = 0, right: int = 0, channel_info=NOTHING, signature: bytes = b'8BIM', blend_mode=<BlendMode.NORMAL: b'norm'>, opacity: int = 255, clipping=<Clipping.BASE: 0>, flags=NOTHING, mask_data=None, blending_ranges=NOTHING, name: str = "", tagged_blocks=NOTHING*)

Layer record.

top

Top position.

left

Left position.

bottom

Bottom position.

right

Right position.

channel_info

List of *ChannelInfo*.

signature

Blend mode signature `b'8BIM'`.

blend_mode

Blend mode key. See *BlendMode*.

opacity
Opacity, 0 = transparent, 255 = opaque.

clipping
Clipping, 0 = base, 1 = non-base. See *Clipping*.

flags
See *LayerFlags*.

mask_data
MaskData or None.

blending_ranges
See *LayerBlendingRanges*.

name
Layer name.

tagged_blocks
See *TaggedBlocks*.

blend_mode

blending_ranges

bottom

channel_info

channel_sizes
List of channel sizes: [(width, height)].

clipping

flags

height
Height of the layer.

left

mask_data

name

opacity

classmethod read (*fp*, *encoding='macroman'*, *version=1*)
Read the element from a file-like object.

Parameters

- **fp** – file-like object
- **encoding** – encoding of the string
- **version** – psd file version

Return type *LayerRecord*

right

signature

tagged_blocks

top

width

Width of the layer.

write (*fp*, *encoding='macroman'*, *version=1*)

Write the element to a file-like object.

Parameters

- **fp** – file-like object
- **encoding** – encoding of the string
- **version** – psd file version

3.16.6 LayerFlags

```
class psd_tools.psd.layer_and_mask.LayerFlags (transparency_protected: bool = False,
visible: bool = True, obsolete: bool = False,
photoshop_v5_later: bool = False,
pixel_data_irrelevant: bool = False,
undocumented_1: bool = False, un-
documented_2: bool = False, undoc-
umented_3: bool = False)
```

Layer flags.

Note there are undocumented flags. Maybe photoshop version.

transparency_protected**visible****pixel_data_irrelevant****obsolete****photoshop_v5_later****pixel_data_irrelevant****classmethod read** (*fp*)

Read the element from a file-like object.

Parameters **fp** – file-like object**Return type** *LayerFlags***transparency_protected****undocumented_1****undocumented_2****undocumented_3****visible****write** (*fp*)

Write the element to a file-like object.

Parameters **fp** – file-like object

3.16.7 LayerBlendingRanges

```
class psd_tools.psd.layer_and_mask.LayerBlendingRanges (composite_ranges=NOTHING,  
chan-  
nel_ranges=NOTHING)
```

Layer blending ranges.

All ranges contain 2 black values followed by 2 white values.

composite_ranges

List of composite gray blend source and destination ranges.

channel_ranges

List of channel source and destination ranges.

channel_ranges

composite_ranges

classmethod *read* (*fp*)

Read the element from a file-like object.

Parameters *fp* – file-like object

Return type *LayerBlendingRanges*

write (*fp*)

Write the element to a file-like object.

Parameters *fp* – file-like object

3.16.8 MaskData

```
class psd_tools.psd.layer_and_mask.MaskData (top: int = 0, left: int = 0, bot-  
tom: int = 0, right: int = 0, back-  
ground_color: int = 0, flags=NOTHING,  
parameters=None, real_flags=None,  
real_background_color=None,  
real_top=None, real_left=None,  
real_bottom=None, real_right=None)
```

Mask data.

Real user mask is a final composite mask of vector and pixel masks.

top

Top position.

left

Left position.

bottom

Bottom position.

right

Right position.

background_color

Default color. 0 or 255.

flags

See *MaskFlags*.

parameters

MaskParameters or None.

real_flags

Real user mask flags. See *MaskFlags*.

real_background_color

Real user mask background. 0 or 255.

real_top

Top position of real user mask.

real_left

Left position of real user mask.

real_bottom

Bottom position of real user mask.

real_right

Right position of real user mask.

background_color**bottom****flags****height**

Height of the mask.

left**parameters****classmethod read** (*fp*)

Read the element from a file-like object.

Parameters *fp* – file-like object

Return type *MaskData* or None

real_background_color**real_bottom****real_flags****real_height**

Height of real user mask.

real_left**real_right****real_top****real_width**

Width of real user mask.

right**top****width**

Width of the mask.

write (*fp*)

Write the element to a file-like object.

Parameters *fp* – file-like object

3.16.9 MaskFlags

class `psd_tools.psd.layer_and_mask.MaskFlags` (*pos_relative_to_layer: bool = False, mask_disabled: bool = False, invert_mask: bool = False, user_mask_from_render: bool = False, parameters_applied: bool = False, undocumented_1: bool = False, undocumented_2: bool = False, undocumented_3: bool = False*)

Mask flags.

pos_relative_to_layer

Position relative to layer.

mask_disabled

Layer mask disabled.

invert_mask

Invert layer mask when blending (Obsolete).

user_mask_from_render

The user mask actually came from rendering other data.

parameters_applied

The user and/or vector masks have parameters applied to them.

invert_mask

mask_disabled

parameters_applied

pos_relative_to_layer

classmethod read (*fp*)

Read the element from a file-like object.

Parameters *fp* – file-like object

Return type *MaskFlags*

undocumented_1

undocumented_2

undocumented_3

user_mask_from_render

write (*fp*)

Write the element to a file-like object.

Parameters *fp* – file-like object

3.16.10 MaskParameters

```
class psd_tools.psd.layer_and_mask.MaskParameters (user_mask_density=None,
                                                    user_mask_feather=None, vec-
                                                    tor_mask_density=None, vec-
                                                    tor_mask_feather=None)
```

Mask parameters.

user_mask_density

user_mask_feather

vector_mask_density

vector_mask_feather

classmethod read (*fp*)

Read the element from a file-like object.

Parameters **fp** – file-like object

Return type *MaskParameters*

user_mask_density

user_mask_feather

vector_mask_density

vector_mask_feather

write (*fp*)

Write the element to a file-like object.

Parameters **fp** – file-like object

3.16.11 ChannelInfo

```
class psd_tools.psd.layer_and_mask.ChannelInfo (id=<ChannelID.CHANNEL_0: 0>,
                                                    length: int = 0)
```

Channel information.

id

Channel ID: 0 = red, 1 = green, etc.; -1 = transparency mask; -2 = user supplied layer mask, -3 real user supplied layer mask (when both a user mask and a vector mask are present). See *ChannelID*.

length

Length of the corresponding channel data.

id

length

classmethod read (*fp, version=1*)

Read the element from a file-like object.

Parameters

- **fp** – file-like object
- **version** – psd file version

Return type *ChannelInfo*

write (*fp*, *version=1*)
Write the element to a file-like object.

Parameters

- **fp** – file-like object
- **version** – psd file version

3.16.12 ChannelImageData

class `psd_tools.psd.layer_and_mask.ChannelImageData` (*items=NOTHING*)
List of channel image data.

See *ChannelDataList*.

classmethod read (*fp*, *layer_records=None*)
Read the element from a file-like object.

Parameters **fp** – file-like object

Return type *ChannelImageData*

write (*fp*, ***kwargs*)
Write the element to a file-like object.

Parameters **fp** – file-like object

3.16.13 ChannelDataList

class `psd_tools.psd.layer_and_mask.ChannelDataList` (*items=NOTHING*)
List of channel image data.

See *ChannelData*.

classmethod read (*fp*, *channel_info*)
Read the element from a file-like object.

Parameters

- **fp** – file-like object
- **channel_info** – *ChannelInfo*

Return type *ChannelDataList*

3.16.14 ChannelData

class `psd_tools.psd.layer_and_mask.ChannelData` (*compression=<Compression.RAW: 0>*,
data: bytes = b'')

Channel data.

compression
Compression type. See *Compression*.

data
Data.

compression

data

get_data (*width, height, depth, version=1*)
Get decompressed channel data.

Parameters

- **width** – width.
- **height** – height.
- **depth** – bit depth of the pixel.
- **version** – psd file version.

Return type bytes

classmethod read (*fp*)
Read the element from a file-like object.

Parameters **fp** – file-like object

Return type *ChannelData*

set_data (*data, width, height, depth, version=1*)
Set raw channel data and compress to store.

Parameters

- **data** – raw data bytes to write.
- **compression** – compression type, see *Compression*.
- **width** – width.
- **height** – height.
- **depth** – bit depth of the pixel.
- **version** – psd file version.

write (*fp, **kwargs*)
Write the element to a file-like object.

Parameters **fp** – file-like object

3.17 psd_tools.psd.tagged_blocks

Tagged block data structure.

Todo: Support the following tagged blocks: TaggedBlockID.PATTERN_DATA, TaggedBlockID.TYPE_TOOL_INFO, TaggedBlockID.LAYER, TaggedBlockID.ALPHA

3.17.1 TaggedBlocks

class psd_tools.psd.tagged_blocks.**TaggedBlocks** (*items=NOTHING*)
Dict of tagged blocks.

Example:

```

from psd_tools.constants import TaggedBlockID

# Iterate over fields
for key in tagged_blocks:
    print(key)

# Get a field
block = tagged_blocks.get(TaggedBlockID.TYPE_TOOL_OBJECT_SETTING)
type_setting = block.data

block = tagged_blocks[TaggedBlockID(b'TySh')]

```

3.17.2 TaggedBlock

class `psd_tools.psd.tagged_blocks.TaggedBlock` (*signature=b'8BIM', key=b'', data=b''*)
 Layer tagged block with extra info.

key
 4-character code. See TaggedBlock

data
 Data.

3.18 psd_tools.psd.vector

Vector mask, path, and stroke structure.

3.18.1 Subpath

class `psd_tools.psd.vector.Subpath` (*items=NOTHING, operation: int = 1, unknown1: int = 1, unknown2: int = 0, index: int = 0, unknown3: bytes = b'x00x00x00x00x00x00x00x00x00'*)

Subpath element.

Operation types:

1: Or (union), 2: Not-Or, 3: And (intersect), 4: Xor (exclude), -1: Subtract?

index

operation

classmethod read (*fp*)

Read the element from a file-like object.

Parameters *fp* – file-like object

write (*fp*)

Write the element to a file-like object.

3.18.2 Knot

class `psd_tools.psd.vector.Knot` (*preceding: tuple = (0.0, 0.0), anchor: tuple = (0.0, 0.0), leaving: tuple = (0.0, 0.0)*)

Knot element consisting of 3 control point for Bezier curves.

..py:attribute: preceding ..py:attribute: anchor ..py:attribute: leaving

anchor

leaving

preceding

classmethod read (*fp*)

Read the element from a file-like object.

Parameters *fp* – file-like object

write (*fp*)

Write the element to a file-like object.

3.18.3 VectorMaskSetting

class `psd_tools.psd.vector.VectorMaskSetting` (*version: int = 3, flags: int = 0, path=None*)
VectorMaskSetting structure.

version

invert

not_link

disable

path

3.18.4 VectorStrokeContentSetting

class `psd_tools.psd.vector.VectorStrokeContentSetting` (*items=NOTHING,*
name: str = "", classID=<DescriptorClassID.NULL:
b'null'>, key: bytes =
b'x00x00x00x00', version: int
= 1)

Dict-like Descriptor-based structure. See *Descriptor*.

key

version

CHAPTER 4

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

p

`psd_tools.api.adjustments`, 15
`psd_tools.api.effects`, 27
`psd_tools.api.layers`, 35
`psd_tools.api.mask`, 50
`psd_tools.api.shape`, 51
`psd_tools.api.smart_object`, 56
`psd_tools.constants`, 56
`psd_tools.psd`, 67
`psd_tools.psd.color_mode_data`, 68
`psd_tools.psd.descriptor`, 68
`psd_tools.psd.engine_data`, 76
`psd_tools.psd.header`, 79
`psd_tools.psd.image_data`, 80
`psd_tools.psd.image_resources`, 81
`psd_tools.psd.layer_and_mask`, 83
`psd_tools.psd.tagged_blocks`, 93
`psd_tools.psd.vector`, 94

A

- Alias (class in psd_tools.psd.descriptor), 69
- ALIAS (psd_tools.constants.LinkedLayerType attribute), 59
- aligned (psd_tools.api.effects.GradientOverlay attribute), 31
- aligned (psd_tools.api.effects.PatternOverlay attribute), 32
- ALPHA (psd_tools.constants.TaggedBlockID attribute), 64
- ALPHA_IDENTIFIERS (psd_tools.constants.ImageResourceID attribute), 60
- ALPHA_NAMES_PASCAL (psd_tools.constants.ImageResourceID attribute), 60
- ALPHA_NAMES_UNICODE (psd_tools.constants.ImageResourceID attribute), 60
- ALTERNATE_DUOTONE_COLORS (psd_tools.constants.ImageResourceID attribute), 60
- ALTERNATE_SPOT_COLORS (psd_tools.constants.ImageResourceID attribute), 60
- altitude (psd_tools.api.effects.BevelEmboss attribute), 33
- anchor (psd_tools.psd.vector.Knot attribute), 95
- angle (psd_tools.api.effects.BevelEmboss attribute), 33
- angle (psd_tools.api.effects.DropShadow attribute), 28
- angle (psd_tools.api.effects.GradientOverlay attribute), 31
- angle (psd_tools.api.effects.InnerShadow attribute), 29
- angle (psd_tools.api.effects.Satin attribute), 34
- ANIMATION_EFFECTS (psd_tools.constants.TaggedBlockID attribute), 64
- ANNOTATIONS (psd_tools.constants.TaggedBlockID attribute), 64
- anti_aliased (psd_tools.api.effects.BevelEmboss attribute), 33
- anti_aliased (psd_tools.api.effects.DropShadow attribute), 28
- anti_aliased (psd_tools.api.effects.InnerGlow attribute), 30
- anti_aliased (psd_tools.api.effects.InnerShadow attribute), 29
- anti_aliased (psd_tools.api.effects.OuterGlow attribute), 29
- anti_aliased (psd_tools.api.effects.Satin attribute), 35
- arrow_conc (psd_tools.api.shape.Line attribute), 53
- arrow_end (psd_tools.api.shape.Line attribute), 53
- arrow_length (psd_tools.api.shape.Line attribute), 53
- arrow_start (psd_tools.api.shape.Line attribute), 53
- arrow_width (psd_tools.api.shape.Line attribute), 53
- ARTBOARD_DATA1 (psd_tools.constants.TaggedBlockID attribute), 64
- ARTBOARD_DATA2 (psd_tools.constants.TaggedBlockID attribute), 64
- ARTBOARD_DATA3 (psd_tools.constants.TaggedBlockID attribute), 64
- AUTO_SAVE_FILE_PATH (psd_tools.constants.ImageResourceID attribute), 60
- AUTO_SAVE_FORMAT (psd_tools.constants.ImageResourceID attribute), 60
- automatic (psd_tools.api.adjustments.BrightnessContrast attribute), 24

B

- background_color (psd_tools.api.mask.Mask attribute), 50
- BACKGROUND_COLOR (psd_tools.constants.ImageResourceID attribute), 60
- background_color (psd_tools.psd.layer_and_mask.MaskData attribute), 88, 89
- BASE (psd_tools.constants.Clipping attribute), 58
- bbox (psd_tools.api.adjustments.GradientFill attribute), 21

- bbox (psd_tools.api.adjustments.PatternFill attribute), 18
- bbox (psd_tools.api.adjustments.SolidColorFill attribute), 16
- bbox (psd_tools.api.layers.Group attribute), 35
- bbox (psd_tools.api.layers.PixelLayer attribute), 38
- bbox (psd_tools.api.layers.ShapeLayer attribute), 41
- bbox (psd_tools.api.layers.SmartObjectLayer attribute), 44
- bbox (psd_tools.api.layers.TypeLayer attribute), 47
- bbox (psd_tools.api.mask.Mask attribute), 50
- bbox (psd_tools.api.shape.Ellipse attribute), 54
- bbox (psd_tools.api.shape.Line attribute), 53
- bbox (psd_tools.api.shape.Rectangle attribute), 54
- bbox (psd_tools.api.shape.RoundedRectangle attribute), 55
- bbox (psd_tools.api.shape.VectorMask attribute), 51
- bbox (psd_tools.PSDImage attribute), 12
- BEVEL (psd_tools.constants.EffectOSType attribute), 59
- bevel_style (psd_tools.api.effects.BevelEmboss attribute), 33
- bevel_type (psd_tools.api.effects.BevelEmboss attribute), 33
- BevelEmboss (class in psd_tools.api.effects), 33
- BITMAP (psd_tools.constants.ColorMode attribute), 58
- BLACK_AND_WHITE (psd_tools.constants.TaggedBlockID attribute), 64
- BlackAndWhite (class in psd_tools.api.adjustments), 26
- BLEND_CLIPPING_ELEMENTS (psd_tools.constants.TaggedBlockID attribute), 64
- BLEND_FILL_OPACITY (psd_tools.constants.TaggedBlockID attribute), 64
- BLEND_INTERIOR_ELEMENTS (psd_tools.constants.TaggedBlockID attribute), 64
- blend_mode (psd_tools.api.adjustments.GradientFill attribute), 21
- blend_mode (psd_tools.api.adjustments.PatternFill attribute), 18
- blend_mode (psd_tools.api.adjustments.SolidColorFill attribute), 16
- blend_mode (psd_tools.api.effects.ColorOverlay attribute), 31
- blend_mode (psd_tools.api.effects.DropShadow attribute), 28
- blend_mode (psd_tools.api.effects.GradientOverlay attribute), 31
- blend_mode (psd_tools.api.effects.InnerGlow attribute), 30
- blend_mode (psd_tools.api.effects.InnerShadow attribute), 29
- blend_mode (psd_tools.api.effects.OuterGlow attribute), 29
- blend_mode (psd_tools.api.effects.PatternOverlay attribute), 32
- blend_mode (psd_tools.api.effects.Satin attribute), 35
- blend_mode (psd_tools.api.effects.Stroke attribute), 33
- blend_mode (psd_tools.api.layers.Group attribute), 35
- blend_mode (psd_tools.api.layers.PixelLayer attribute), 38
- blend_mode (psd_tools.api.layers.ShapeLayer attribute), 41
- blend_mode (psd_tools.api.layers.SmartObjectLayer attribute), 44
- blend_mode (psd_tools.api.layers.TypeLayer attribute), 47
- blend_mode (psd_tools.api.shape.Stroke attribute), 52
- blend_mode (psd_tools.psd.layer_and_mask.LayerRecord attribute), 85, 86
- blending_ranges (psd_tools.psd.layer_and_mask.LayerRecord attribute), 86
- BlendMode (class in psd_tools.constants), 56
- blue (psd_tools.api.adjustments.BlackAndWhite attribute), 26
- Bool (class in psd_tools.psd.descriptor), 69
- Bool (class in psd_tools.psd.engine_data), 76
- BORDER_INFO (psd_tools.constants.ImageResourceID attribute), 60
- bottom (psd_tools.api.adjustments.GradientFill attribute), 21
- bottom (psd_tools.api.adjustments.PatternFill attribute), 19
- bottom (psd_tools.api.adjustments.SolidColorFill attribute), 16
- bottom (psd_tools.api.layers.PixelLayer attribute), 38
- bottom (psd_tools.api.layers.ShapeLayer attribute), 41
- bottom (psd_tools.api.layers.SmartObjectLayer attribute), 44
- bottom (psd_tools.api.layers.TypeLayer attribute), 47
- bottom (psd_tools.api.mask.Mask attribute), 50
- bottom (psd_tools.psd.layer_and_mask.LayerRecord attribute), 85, 86
- bottom (psd_tools.psd.layer_and_mask.MaskData attribute), 88, 89
- bottom (psd_tools.PSDImage attribute), 12
- BOUNDING_SECTION_DIVIDER (psd_tools.constants.SectionDivider attribute), 64
- brightness (psd_tools.api.adjustments.BrightnessContrast attribute), 24
- BRIGHTNESS_AND_CONTRAST (psd_tools.constants.TaggedBlockID attribute), 64
- BrightnessContrast (class in psd_tools.api.adjustments), 24

C

- CAPTION_DIGEST (psd_tools.constants.ImageResourceID attribute), 60
- CAPTION_PASCAL (psd_tools.constants.ImageResourceID attribute), 60
- CENTERED (psd_tools.constants.PrintScaleStyle attribute), 64
- CHANNEL_0 (psd_tools.constants.ChannelID attribute), 57
- CHANNEL_1 (psd_tools.constants.ChannelID attribute), 57
- CHANNEL_2 (psd_tools.constants.ChannelID attribute), 57
- CHANNEL_3 (psd_tools.constants.ChannelID attribute), 57
- CHANNEL_4 (psd_tools.constants.ChannelID attribute), 57
- CHANNEL_5 (psd_tools.constants.ChannelID attribute), 57
- CHANNEL_6 (psd_tools.constants.ChannelID attribute), 58
- CHANNEL_7 (psd_tools.constants.ChannelID attribute), 58
- CHANNEL_8 (psd_tools.constants.ChannelID attribute), 58
- CHANNEL_9 (psd_tools.constants.ChannelID attribute), 58
- CHANNEL_BLENDING_RESTRICTIONS_SETTING (psd_tools.constants.TaggedBlockID attribute), 65
- channel_image_data (psd_tools.psd.layer_and_mask.LayerInfo attribute), 84
- channel_info (psd_tools.psd.layer_and_mask.LayerRecord attribute), 85, 86
- CHANNEL_MIXER (psd_tools.constants.TaggedBlockID attribute), 65
- channel_ranges (psd_tools.psd.layer_and_mask.LayerBlendingRanges attribute), 88
- channel_sizes (psd_tools.psd.layer_and_mask.LayerRecord attribute), 86
- ChannelData (class in psd_tools.psd.layer_and_mask), 92
- ChannelDataList (class in psd_tools.psd.layer_and_mask), 92
- ChannelID (class in psd_tools.constants), 57
- ChannelImageData (class in psd_tools.psd.layer_and_mask), 92
- ChannelInfo (class in psd_tools.psd.layer_and_mask), 91
- ChannelMixer (class in psd_tools.api.adjustments), 26
- channels (psd_tools.psd.header.FileHeader attribute), 79
- channels (psd_tools.PSDImage attribute), 12
- choke (psd_tools.api.effects.DropShadow attribute), 28
- choke (psd_tools.api.effects.InnerGlow attribute), 30
- choke (psd_tools.api.effects.InnerShadow attribute), 29
- choke (psd_tools.api.effects.OuterGlow attribute), 29
- Class (class in psd_tools.psd.descriptor), 69
- Class1 (class in psd_tools.psd.descriptor), 69
- Class2 (class in psd_tools.psd.descriptor), 70
- Class3 (class in psd_tools.psd.descriptor), 70
- classID (psd_tools.psd.descriptor.Class attribute), 69
- classID (psd_tools.psd.descriptor.Descriptor attribute), 70
- classID (psd_tools.psd.descriptor.ObjectArray attribute), 73
- clip_layers (psd_tools.api.adjustments.GradientFill attribute), 21
- clip_layers (psd_tools.api.adjustments.PatternFill attribute), 19
- clip_layers (psd_tools.api.adjustments.SolidColorFill attribute), 16
- clip_layers (psd_tools.api.layers.Group attribute), 36
- clip_layers (psd_tools.api.layers.PixelLayer attribute), 38
- clip_layers (psd_tools.api.layers.ShapeLayer attribute), 41
- clip_layers (psd_tools.api.layers.SmartObjectLayer attribute), 44
- clip_layers (psd_tools.api.layers.TypeLayer attribute), 47
- CLIPBOARD (psd_tools.constants.PathResourceID attribute), 63
- clipboard_record (psd_tools.api.shape.VectorMask attribute), 51
- Clipping (class in psd_tools.constants), 58
- clipping (psd_tools.psd.layer_and_mask.LayerRecord attribute), 86
- CLIPPING_PATH_NAME (psd_tools.constants.ImageResourceID attribute), 60
- CLOSED_FOLDER (psd_tools.constants.SectionDivider attribute), 64
- CLOSED_KNOT_LINKED (psd_tools.constants.PathResourceID attribute), 63
- CLOSED_KNOT_UNLINKED (psd_tools.constants.PathResourceID attribute), 63
- CLOSED_LENGTH (psd_tools.constants.PathResourceID attribute), 63
- CMYK (psd_tools.constants.ColorMode attribute), 58
- CMYK (psd_tools.constants.ColorSpaceID attribute), 58
- color (psd_tools.api.effects.ColorOverlay attribute), 31
- color (psd_tools.api.effects.DropShadow attribute), 28
- color (psd_tools.api.effects.InnerGlow attribute), 30
- color (psd_tools.api.effects.InnerShadow attribute), 29
- color (psd_tools.api.effects.OuterGlow attribute), 29
- color (psd_tools.api.effects.Satin attribute), 35
- color (psd_tools.api.effects.Stroke attribute), 33
- COLOR (psd_tools.constants.BlendMode attribute), 56
- COLOR_BALANCE (psd_tools.constants.TaggedBlockID attribute), 65
- COLOR_BURN (psd_tools.constants.BlendMode attribute), 56

- tribute), 56
 - color_components (psd_tools.api.adjustments.PhotoFilter attribute), 26
 - COLOR_DODGE (psd_tools.constants.BlendMode attribute), 57
 - COLOR_HALFTONING_INFO (psd_tools.constants.ImageResourceID attribute), 60
 - COLOR_LOOKUP (psd_tools.constants.TaggedBlockID attribute), 65
 - color_mode (psd_tools.psd.header.FileHeader attribute), 79
 - color_mode (psd_tools.PSDImage attribute), 12
 - color_mode_data (psd_tools.psd.PSD attribute), 67
 - color_model (psd_tools.api.adjustments.GradientMap attribute), 27
 - COLOR_PROTECTED (psd_tools.constants.GlobalLayerMaskKind attribute), 59
 - COLOR_SAMPLERS_RESOURCE (psd_tools.constants.ImageResourceID attribute), 60
 - COLOR_SAMPLERS_RESOURCE_OBSOLETE (psd_tools.constants.ImageResourceID attribute), 60
 - COLOR_SELECTED (psd_tools.constants.GlobalLayerMaskKind attribute), 59
 - color_space (psd_tools.api.adjustments.PhotoFilter attribute), 26
 - color_stops (psd_tools.api.adjustments.GradientMap attribute), 27
 - COLOR_TRANSFER_FUNCTION (psd_tools.constants.ImageResourceID attribute), 60
 - ColorBalance (class in psd_tools.api.adjustments), 25
 - colorization (psd_tools.api.adjustments.HueSaturation attribute), 25
 - ColorLookup (class in psd_tools.api.adjustments), 26
 - ColorMode (class in psd_tools.constants), 58
 - ColorModeData (class in psd_tools.psd.color_mode_data), 68
 - ColorOverlay (class in psd_tools.api.effects), 31
 - ColorSpaceID (class in psd_tools.constants), 58
 - COMMON_STATE (psd_tools.constants.EffectOSType attribute), 59
 - compose() (in module psd_tools), 15
 - compose() (psd_tools.api.adjustments.GradientFill method), 21
 - compose() (psd_tools.api.adjustments.PatternFill method), 19
 - compose() (psd_tools.api.adjustments.SolidColorFill method), 16
 - compose() (psd_tools.api.layers.Group method), 36
 - compose() (psd_tools.api.layers.Pixellayer method), 39
 - compose() (psd_tools.api.layers.ShapeLayer method), 41
 - compose() (psd_tools.api.layers.SmartObjectLayer method), 44
 - compose() (psd_tools.api.layers.TypeLayer method), 47
 - compose() (psd_tools.PSDImage method), 12
 - composite_ranges (psd_tools.psd.layer_and_mask.LayerBlendingRanges attribute), 88
 - Compression (class in psd_tools.constants), 59
 - compression (psd_tools.psd.image_data.ImageData attribute), 80
 - compression (psd_tools.psd.layer_and_mask.ChannelData attribute), 92
 - COMPUTER_INFO (psd_tools.constants.TaggedBlockID attribute), 65
 - content (psd_tools.api.shape.Stroke attribute), 52
 - CONTENT_GENERATOR_EXTRA_DATA (psd_tools.constants.TaggedBlockID attribute), 65
 - contour (psd_tools.api.effects.BevelEmboss attribute), 34
 - contour (psd_tools.api.effects.DropShadow attribute), 28
 - contour (psd_tools.api.effects.InnerGlow attribute), 30
 - contour (psd_tools.api.effects.InnerShadow attribute), 29
 - contour (psd_tools.api.effects.OuterGlow attribute), 30
 - contour (psd_tools.api.effects.Satin attribute), 35
 - contrast (psd_tools.api.adjustments.BrightnessContrast attribute), 24
 - COPYRIGHT_FLAG (psd_tools.constants.ImageResourceID attribute), 60
 - COUNT_INFO (psd_tools.constants.ImageResourceID attribute), 60
 - Curves (class in psd_tools.api.adjustments), 24
 - CURVES (psd_tools.constants.TaggedBlockID attribute), 65
 - cyan (psd_tools.api.adjustments.BlackAndWhite attribute), 26
- ## D
- DARKEN (psd_tools.constants.BlendMode attribute), 57
 - DARKER_COLOR (psd_tools.constants.BlendMode attribute), 57
 - data (psd_tools.api.adjustments.ChannelMixer attribute), 26
 - data (psd_tools.api.adjustments.Curves attribute), 24
 - data (psd_tools.api.adjustments.GradientFill attribute), 22
 - data (psd_tools.api.adjustments.HueSaturation attribute), 25
 - data (psd_tools.api.adjustments.Levels attribute), 25
 - data (psd_tools.api.adjustments.PatternFill attribute), 19
 - data (psd_tools.api.adjustments.SelectiveColor attribute), 27
 - data (psd_tools.api.adjustments.SolidColorFill attribute), 16
 - data (psd_tools.api.smart_object.SmartObject attribute), 56

- DATA (psd_tools.constants.LinkedLayerType attribute), 59
- data (psd_tools.psd.image_data.ImageData attribute), 80
- data (psd_tools.psd.image_resources.ImageResource attribute), 82
- data (psd_tools.psd.layer_and_mask.ChannelData attribute), 92
- data (psd_tools.psd.tagged_blocks.TaggedBlock attribute), 94
- density (psd_tools.api.adjustments.PhotoFilter attribute), 26
- depth (psd_tools.api.effects.BevelEmboss attribute), 34
- depth (psd_tools.psd.header.FileHeader attribute), 79
- depth (psd_tools.PSDImage attribute), 12
- descendants() (psd_tools.api.layers.Group method), 36
- descendants() (psd_tools.PSDImage method), 12
- Descriptor (class in psd_tools.psd.descriptor), 70
- Dict (class in psd_tools.psd.engine_data), 77
- DIFFERENCE (psd_tools.constants.BlendMode attribute), 57
- direction (psd_tools.api.effects.BevelEmboss attribute), 34
- disable (psd_tools.psd.vector.VectorMaskSetting attribute), 95
- disabled (psd_tools.api.mask.Mask attribute), 50
- disabled (psd_tools.api.shape.VectorMask attribute), 51
- DISPLAY_INFO (psd_tools.constants.ImageResourceID attribute), 60
- DISPLAY_INFO_OBSOLETE (psd_tools.constants.ImageResourceID attribute), 60
- DISSOLVE (psd_tools.constants.BlendMode attribute), 57
- distance (psd_tools.api.effects.DropShadow attribute), 28
- distance (psd_tools.api.effects.InnerShadow attribute), 29
- distance (psd_tools.api.effects.Satin attribute), 35
- dithered (psd_tools.api.adjustments.GradientMap attribute), 27
- dithered (psd_tools.api.effects.GradientOverlay attribute), 32
- DIVIDE (psd_tools.constants.BlendMode attribute), 57
- document_resources (psd_tools.api.layers.TypeLayer attribute), 47
- Double (class in psd_tools.psd.descriptor), 70
- DROP_SHADOW (psd_tools.constants.EffectOSType attribute), 59
- DropShadow (class in psd_tools.api.effects), 28
- DUOTONE (psd_tools.constants.ColorMode attribute), 58
- DUOTONE_HALFTONING_INFO (psd_tools.constants.ImageResourceID attribute), 60
- DUOTONE_IMAGE_INFO (psd_tools.constants.ImageResourceID attribute), 60
- DUOTONE_TRANSFER_FUNCTION (psd_tools.constants.ImageResourceID attribute), 60
- ## E
- EFFECTIVE_BW (psd_tools.constants.ImageResourceID attribute), 60
- EffectOSType (class in psd_tools.constants), 59
- Effects (class in psd_tools.api.effects), 27
- effects (psd_tools.api.adjustments.GradientFill attribute), 22
- effects (psd_tools.api.adjustments.PatternFill attribute), 19
- effects (psd_tools.api.adjustments.SolidColorFill attribute), 16
- effects (psd_tools.api.layers.Group attribute), 36
- effects (psd_tools.api.layers.PixelLayer attribute), 39
- effects (psd_tools.api.layers.ShapeLayer attribute), 41
- effects (psd_tools.api.layers.SmartObjectLayer attribute), 44
- effects (psd_tools.api.layers.TypeLayer attribute), 47
- EFFECTS_LAYER (psd_tools.constants.TaggedBlockID attribute), 65
- EFFECTS_VISIBLE (psd_tools.constants.ImageResourceID attribute), 60
- Ellipse (class in psd_tools.api.shape), 54
- enable_colorization (psd_tools.api.adjustments.HueSaturation attribute), 25
- enabled (psd_tools.api.effects.BevelEmboss attribute), 34
- enabled (psd_tools.api.effects.ColorOverlay attribute), 31
- enabled (psd_tools.api.effects.DropShadow attribute), 28
- enabled (psd_tools.api.effects.Effects attribute), 28
- enabled (psd_tools.api.effects.GradientOverlay attribute), 32
- enabled (psd_tools.api.effects.InnerGlow attribute), 30
- enabled (psd_tools.api.effects.InnerShadow attribute), 29
- enabled (psd_tools.api.effects.OuterGlow attribute), 30
- enabled (psd_tools.api.effects.PatternOverlay attribute), 32
- enabled (psd_tools.api.effects.Satin attribute), 35
- enabled (psd_tools.api.effects.Stroke attribute), 33
- enabled (psd_tools.api.shape.Stroke attribute), 52
- engine_dict (psd_tools.api.layers.TypeLayer attribute), 47
- EngineData (class in psd_tools.psd.engine_data), 76
- EngineData2 (class in psd_tools.psd.engine_data), 76
- Enum (class in psd_tools.psd.descriptor), 71
- enum (psd_tools.psd.image_resources.ImageResources attribute), 82
- EnumeratedReference (class in psd_tools.psd.descriptor), 71
- EPS_OPTIONS (psd_tools.constants.ImageResourceID attribute), 60

EXCLUSION (psd_tools.constants.BlendMode attribute), 57

EXIF_DATA_1 (psd_tools.constants.ImageResourceID attribute), 60

EXIF_DATA_3 (psd_tools.constants.ImageResourceID attribute), 60

expansion (psd_tools.api.adjustments.GradientMap attribute), 27

EXPORT_SETTING1 (psd_tools.constants.TaggedBlockID attribute), 65

EXPORT_SETTING2 (psd_tools.constants.TaggedBlockID attribute), 65

Exposure (class in psd_tools.api.adjustments), 24

exposure (psd_tools.api.adjustments.Exposure attribute), 24

EXPOSURE (psd_tools.constants.TaggedBlockID attribute), 65

EXTERNAL (psd_tools.constants.LinkedLayerType attribute), 59

extra (psd_tools.api.adjustments.Curves attribute), 24

F

FileHeader (class in psd_tools.psd.header), 79

filename (psd_tools.api.smart_object.SmartObject attribute), 56

filesize (psd_tools.api.smart_object.SmartObject attribute), 56

filetype (psd_tools.api.smart_object.SmartObject attribute), 56

fill_enabled (psd_tools.api.shape.Stroke attribute), 52

fill_type (psd_tools.api.effects.Stroke attribute), 33

FILTER_EFFECTS1 (psd_tools.constants.TaggedBlockID attribute), 65

FILTER_EFFECTS2 (psd_tools.constants.TaggedBlockID attribute), 65

FILTER_EFFECTS3 (psd_tools.constants.TaggedBlockID attribute), 65

FILTER_MASK (psd_tools.constants.TaggedBlockID attribute), 65

flags (psd_tools.api.mask.Mask attribute), 50

flags (psd_tools.psd.layer_and_mask.LayerRecord attribute), 86

flags (psd_tools.psd.layer_and_mask.MaskData attribute), 88, 89

Float (class in psd_tools.psd.engine_data), 77

FOREIGN_EFFECT_ID (psd_tools.constants.TaggedBlockID attribute), 65

frombytes() (psd_tools.psd.engine_data.Bool class method), 76

frombytes() (psd_tools.psd.engine_data.Dict class method), 77

frombytes() (psd_tools.psd.engine_data.Float class method), 77

frombytes() (psd_tools.psd.engine_data.Integer class method), 77

frombytes() (psd_tools.psd.engine_data.List class method), 78

frombytes() (psd_tools.psd.engine_data.Property class method), 78

frombytes() (psd_tools.psd.engine_data.String class method), 78

frombytes() (psd_tools.psd.PSD class method), 67

frompil() (psd_tools.PSDImage class method), 12

G

gamma (psd_tools.api.adjustments.Exposure attribute), 25

get_data() (psd_tools.psd.image_data.ImageData method), 80

get_data() (psd_tools.psd.image_resources.ImageResources method), 82

get_data() (psd_tools.psd.layer_and_mask.ChannelData method), 92

GLOBAL_ALTITUDE (psd_tools.constants.ImageResourceID attribute), 60

GLOBAL_ANGLE (psd_tools.constants.ImageResourceID attribute), 60

global_layer_mask_info (psd_tools.psd.layer_and_mask.LayerAndMaskInfo attribute), 83

GlobalLayerMaskInfo (class in psd_tools.psd.layer_and_mask), 84

GlobalLayerMaskKind (class in psd_tools.constants), 59

GlobalObject (class in psd_tools.psd.descriptor), 71

glow_source (psd_tools.api.effects.InnerGlow attribute), 30

glow_type (psd_tools.api.effects.InnerGlow attribute), 30

glow_type (psd_tools.api.effects.OuterGlow attribute), 30

gradient (psd_tools.api.effects.GradientOverlay attribute), 32

gradient (psd_tools.api.effects.InnerGlow attribute), 31

gradient (psd_tools.api.effects.OuterGlow attribute), 30

gradient (psd_tools.api.effects.Stroke attribute), 33

GRADIENT_FILL_SETTING (psd_tools.constants.TaggedBlockID attribute), 65

gradient_kind (psd_tools.api.adjustments.GradientFill attribute), 22

GRADIENT_MAP (psd_tools.constants.TaggedBlockID attribute), 65

gradient_name (psd_tools.api.adjustments.GradientMap attribute), 27

GradientFill (class in psd_tools.api.adjustments), 21

GradientMap (class in psd_tools.api.adjustments), 27

GradientOverlay (class in psd_tools.api.effects), 31

GRAYSCALE (psd_tools.constants.ColorMode attribute), 58

GRAYSCALE (psd_tools.constants.ColorSpaceID attribute), 58
 GRAYSCALE_HALFTONING_INFO (psd_tools.constants.ImageResourceID attribute), 60
 GRAYSCALE_TRANSFER_FUNCTION (psd_tools.constants.ImageResourceID attribute), 61
 green (psd_tools.api.adjustments.BlackAndWhite attribute), 26
 GRID_AND_GUIDES_INFO (psd_tools.constants.ImageResourceID attribute), 61
 Group (class in psd_tools.api.layers), 35

H

HARD_LIGHT (psd_tools.constants.BlendMode attribute), 57
 HARD_MIX (psd_tools.constants.BlendMode attribute), 57
 has_clip_layers() (psd_tools.api.adjustments.GradientFill method), 22
 has_clip_layers() (psd_tools.api.adjustments.PatternFill method), 19
 has_clip_layers() (psd_tools.api.adjustments.SolidColorFill method), 16
 has_clip_layers() (psd_tools.api.layers.Group method), 36
 has_clip_layers() (psd_tools.api.layers.PixelLayer method), 39
 has_clip_layers() (psd_tools.api.layers.ShapeLayer method), 41
 has_clip_layers() (psd_tools.api.layers.SmartObjectLayer method), 44
 has_clip_layers() (psd_tools.api.layers.TypeLayer method), 48
 has_effects() (psd_tools.api.adjustments.GradientFill method), 22
 has_effects() (psd_tools.api.adjustments.PatternFill method), 19
 has_effects() (psd_tools.api.adjustments.SolidColorFill method), 16
 has_effects() (psd_tools.api.layers.Group method), 36
 has_effects() (psd_tools.api.layers.PixelLayer method), 39
 has_effects() (psd_tools.api.layers.ShapeLayer method), 41
 has_effects() (psd_tools.api.layers.SmartObjectLayer method), 44
 has_effects() (psd_tools.api.layers.TypeLayer method), 48
 has_mask() (psd_tools.api.adjustments.GradientFill method), 22
 has_mask() (psd_tools.api.adjustments.PatternFill method), 19
 has_mask() (psd_tools.api.adjustments.SolidColorFill method), 17
 has_mask() (psd_tools.api.layers.Group method), 36
 has_mask() (psd_tools.api.layers.PixelLayer method), 39
 has_mask() (psd_tools.api.layers.ShapeLayer method), 42
 has_mask() (psd_tools.api.layers.SmartObjectLayer method), 44
 has_mask() (psd_tools.api.layers.TypeLayer method), 48
 has_origination() (psd_tools.api.adjustments.GradientFill method), 22
 has_origination() (psd_tools.api.adjustments.PatternFill method), 19
 has_origination() (psd_tools.api.adjustments.SolidColorFill method), 17
 has_origination() (psd_tools.api.layers.Group method), 36
 has_origination() (psd_tools.api.layers.PixelLayer method), 39
 has_origination() (psd_tools.api.layers.ShapeLayer method), 42
 has_origination() (psd_tools.api.layers.SmartObjectLayer method), 44
 has_origination() (psd_tools.api.layers.TypeLayer method), 48
 has_pixels() (psd_tools.api.adjustments.GradientFill method), 22
 has_pixels() (psd_tools.api.adjustments.PatternFill method), 19
 has_pixels() (psd_tools.api.adjustments.SolidColorFill method), 17
 has_pixels() (psd_tools.api.layers.Group method), 36
 has_pixels() (psd_tools.api.layers.PixelLayer method), 39
 has_pixels() (psd_tools.api.layers.ShapeLayer method), 42
 has_pixels() (psd_tools.api.layers.SmartObjectLayer method), 45
 has_pixels() (psd_tools.api.layers.TypeLayer method), 48
 has_preview() (psd_tools.PSDImage method), 13
 has_stroke() (psd_tools.api.layers.ShapeLayer method), 42
 has_thumbnail() (psd_tools.PSDImage method), 13
 has_vector_mask() (psd_tools.api.adjustments.GradientFill method), 22
 has_vector_mask() (psd_tools.api.adjustments.PatternFill method), 19
 has_vector_mask() (psd_tools.api.adjustments.SolidColorFill method), 17
 has_vector_mask() (psd_tools.api.layers.Group method), 37
 has_vector_mask() (psd_tools.api.layers.PixelLayer method), 39

has_vector_mask() (psd_tools.api.layers.ShapeLayer method), 42
 has_vector_mask() (psd_tools.api.layers.SmartObjectLayer method), 45
 has_vector_mask() (psd_tools.api.layers.TypeLayer method), 48
 HDR_TONING_INFO (psd_tools.constants.ImageResourceID attribute), 61
 header (psd_tools.psd.PSD attribute), 67
 height (psd_tools.api.adjustments.GradientFill attribute), 22
 height (psd_tools.api.adjustments.PatternFill attribute), 20
 height (psd_tools.api.adjustments.SolidColorFill attribute), 17
 height (psd_tools.api.layers.Group attribute), 37
 height (psd_tools.api.layers.PixelLayer attribute), 39
 height (psd_tools.api.layers.ShapeLayer attribute), 42
 height (psd_tools.api.layers.SmartObjectLayer attribute), 45
 height (psd_tools.api.layers.TypeLayer attribute), 48
 height (psd_tools.api.mask.Mask attribute), 50
 height (psd_tools.psd.header.FileHeader attribute), 79
 height (psd_tools.psd.layer_and_mask.LayerRecord attribute), 86
 height (psd_tools.psd.layer_and_mask.MaskData attribute), 89
 height (psd_tools.PSDImage attribute), 13
 highlight_color (psd_tools.api.effects.BevelEmboss attribute), 34
 highlight_mode (psd_tools.api.effects.BevelEmboss attribute), 34
 highlight_opacity (psd_tools.api.effects.BevelEmboss attribute), 34
 highlights (psd_tools.api.adjustments.ColorBalance attribute), 26
 HSB (psd_tools.constants.ColorSpaceID attribute), 58
 HUE (psd_tools.constants.BlendMode attribute), 57
 HUE_SATURATION (psd_tools.constants.TaggedBlockID attribute), 65
 HUE_SATURATION_V4 (psd_tools.constants.TaggedBlockID attribute), 65
 HueSaturation (class in psd_tools.api.adjustments), 25
I
 ICC_PROFILE (psd_tools.constants.ImageResourceID attribute), 61
 ICC_UNTAGGED_PROFILE (psd_tools.constants.ImageResourceID attribute), 61
 id (psd_tools.psd.layer_and_mask.ChannelInfo attribute), 91
 Identifier (class in psd_tools.psd.descriptor), 71
 IDS_SEED_NUMBER (psd_tools.constants.ImageResourceID attribute), 61
 image_data (psd_tools.psd.PSD attribute), 67
 IMAGE_MODE_RAW (psd_tools.constants.ImageResourceID attribute), 61
 IMAGE_READY_DATA_SETS (psd_tools.constants.ImageResourceID attribute), 61
 IMAGE_READY_VARIABLES (psd_tools.constants.ImageResourceID attribute), 61
 image_resources (psd_tools.psd.PSD attribute), 67
 image_resources (psd_tools.PSDImage attribute), 13
 IMAGE_STACK (psd_tools.constants.PlacedLayerType attribute), 64
 ImageData (class in psd_tools.psd.image_data), 80
 ImageResource (class in psd_tools.psd.image_resources), 82
 ImageResourceID (class in psd_tools.constants), 60
 ImageResources (class in psd_tools.psd.image_resources), 82
 Index (class in psd_tools.psd.descriptor), 72
 index (psd_tools.api.shape.Ellipse attribute), 54
 index (psd_tools.api.shape.Line attribute), 53
 index (psd_tools.api.shape.Rectangle attribute), 54
 index (psd_tools.api.shape.RoundedRectangle attribute), 55
 index (psd_tools.psd.vector.Subpath attribute), 94
 INDEXED (psd_tools.constants.ColorMode attribute), 58
 INDEXED_COLOR_TABLE_COUNT (psd_tools.constants.ImageResourceID attribute), 61
 INITIAL_FILL (psd_tools.constants.PathResourceID attribute), 63
 initial_fill_rule (psd_tools.api.shape.VectorMask attribute), 51
 INNER_GLOW (psd_tools.constants.EffectOSType attribute), 59
 INNER_SHADOW (psd_tools.constants.EffectOSType attribute), 59
 InnerGlow (class in psd_tools.api.effects), 30
 InnerShadow (class in psd_tools.api.effects), 29
 Integer (class in psd_tools.psd.descriptor), 72
 Integer (class in psd_tools.psd.engine_data), 77
 interleave() (psd_tools.psd.color_mode_data.ColorModeData method), 68
 interpolation (psd_tools.api.adjustments.GradientMap attribute), 27
 Invalidated (class in psd_tools.api.shape), 53
 invalidated (psd_tools.api.shape.Ellipse attribute), 54
 invalidated (psd_tools.api.shape.Invalidated attribute), 53
 invalidated (psd_tools.api.shape.Line attribute), 53
 invalidated (psd_tools.api.shape.Rectangle attribute), 55
 invalidated (psd_tools.api.shape.RoundedRectangle attribute), 55

- tribute), 55
 - INVERT (psd_tools.constants.TaggedBlockID attribute), 65
 - invert (psd_tools.psd.vector.VectorMaskSetting attribute), 95
 - invert_mask (psd_tools.psd.layer_and_mask.MaskFlags attribute), 90
 - inverted (psd_tools.api.effects.Satin attribute), 35
 - inverted (psd_tools.api.shape.VectorMask attribute), 51
 - IPTC_NAA (psd_tools.constants.ImageResourceID attribute), 61
 - is_group() (psd_tools.api.adjustments.GradientFill method), 22
 - is_group() (psd_tools.api.adjustments.PatternFill method), 20
 - is_group() (psd_tools.api.adjustments.SolidColorFill method), 17
 - is_group() (psd_tools.api.layers.Group method), 37
 - is_group() (psd_tools.api.layers.PixelLayer method), 39
 - is_group() (psd_tools.api.layers.ShapeLayer method), 42
 - is_group() (psd_tools.api.layers.SmartObjectLayer method), 45
 - is_group() (psd_tools.api.layers.TypeLayer method), 48
 - is_group() (psd_tools.PSDImage method), 13
 - is_psd() (psd_tools.api.smart_object.SmartObject method), 56
 - is_visible() (psd_tools.api.adjustments.GradientFill method), 22
 - is_visible() (psd_tools.api.adjustments.PatternFill method), 20
 - is_visible() (psd_tools.api.adjustments.SolidColorFill method), 17
 - is_visible() (psd_tools.api.layers.Group method), 37
 - is_visible() (psd_tools.api.layers.PixelLayer method), 39
 - is_visible() (psd_tools.api.layers.ShapeLayer method), 42
 - is_visible() (psd_tools.api.layers.SmartObjectLayer method), 45
 - is_visible() (psd_tools.api.layers.TypeLayer method), 48
 - is_visible() (psd_tools.PSDImage method), 13
 - items (psd_tools.psd.descriptor.List attribute), 72
 - items_count (psd_tools.psd.descriptor.ObjectArray attribute), 73
- ## J
- JPEG_QUALITY (psd_tools.constants.ImageResourceID attribute), 61
 - JUMP_TO_XPEP (psd_tools.constants.ImageResourceID attribute), 61
- ## K
- key (psd_tools.psd.image_resources.ImageResource attribute), 82
 - key (psd_tools.psd.tagged_blocks.TaggedBlock attribute), 94
 - key (psd_tools.psd.vector.VectorStrokeContentSetting attribute), 95
 - kind (psd_tools.api.adjustments.GradientFill attribute), 23
 - kind (psd_tools.api.adjustments.PatternFill attribute), 20
 - kind (psd_tools.api.adjustments.SolidColorFill attribute), 17
 - kind (psd_tools.api.layers.Group attribute), 37
 - kind (psd_tools.api.layers.PixelLayer attribute), 39
 - kind (psd_tools.api.layers.ShapeLayer attribute), 42
 - kind (psd_tools.api.layers.SmartObjectLayer attribute), 45
 - kind (psd_tools.api.layers.TypeLayer attribute), 48
 - kind (psd_tools.api.smart_object.SmartObject attribute), 56
 - kind (psd_tools.psd.layer_and_mask.GlobalLayerMaskInfo attribute), 84
 - kind (psd_tools.PSDImage attribute), 13
 - KNOCKOUT_SETTING (psd_tools.constants.TaggedBlockID attribute), 65
 - Knot (class in psd_tools.psd.vector), 94
- ## L
- lab (psd_tools.api.adjustments.BrightnessContrast attribute), 24
 - LAB (psd_tools.constants.ColorMode attribute), 58
 - LAB (psd_tools.constants.ColorSpaceID attribute), 58
 - LargeInteger (class in psd_tools.psd.descriptor), 72
 - LAYER (psd_tools.constants.TaggedBlockID attribute), 65
 - LAYER_16 (psd_tools.constants.TaggedBlockID attribute), 65
 - LAYER_32 (psd_tools.constants.TaggedBlockID attribute), 65
 - layer_and_mask_information (psd_tools.psd.PSD attribute), 67
 - LAYER_COMPS (psd_tools.constants.ImageResourceID attribute), 61
 - layer_count (psd_tools.psd.layer_and_mask.LayerInfo attribute), 84
 - LAYER_GROUP_INFO (psd_tools.constants.ImageResourceID attribute), 61
 - LAYER_GROUPS_ENABLED_ID (psd_tools.constants.ImageResourceID attribute), 61
 - layer_id (psd_tools.api.adjustments.GradientFill attribute), 23
 - layer_id (psd_tools.api.adjustments.PatternFill attribute), 20
 - layer_id (psd_tools.api.adjustments.SolidColorFill attribute), 17
 - layer_id (psd_tools.api.layers.Group attribute), 37
 - layer_id (psd_tools.api.layers.PixelLayer attribute), 40

- layer_id (psd_tools.api.layers.ShapeLayer attribute), 42
 - layer_id (psd_tools.api.layers.SmartObjectLayer attribute), 45
 - layer_id (psd_tools.api.layers.TypeLayer attribute), 48
 - LAYER_ID (psd_tools.constants.TaggedBlockID attribute), 65
 - layer_info (psd_tools.psd.layer_and_mask.LayerAndMaskInformation (psd_tools.constants.ImageResourceID attribute), 83
 - layer_knocks_out (psd_tools.api.effects.DropShadow attribute), 28
 - LAYER_MASK_AS_GLOBAL_MASK (psd_tools.constants.TaggedBlockID attribute), 65
 - LAYER_NAME_SOURCE_SETTING (psd_tools.constants.TaggedBlockID attribute), 65
 - layer_records (psd_tools.psd.layer_and_mask.LayerInfo attribute), 84
 - LAYER_SELECTION_IDS (psd_tools.constants.ImageResourceID attribute), 61
 - LAYER_STATE_INFO (psd_tools.constants.ImageResourceID attribute), 61
 - LAYER_VERSION (psd_tools.constants.TaggedBlockID attribute), 65
 - LayerAndMaskInformation (class in psd_tools.psd.layer_and_mask), 83
 - LayerBlendingRanges (class in psd_tools.psd.layer_and_mask), 88
 - LayerFlags (class in psd_tools.psd.layer_and_mask), 87
 - LayerInfo (class in psd_tools.psd.layer_and_mask), 84
 - LayerRecord (class in psd_tools.psd.layer_and_mask), 85
 - LayerRecords (class in psd_tools.psd.layer_and_mask), 85
 - leaving (psd_tools.psd.vector.Knot attribute), 95
 - left (psd_tools.api.adjustments.GradientFill attribute), 23
 - left (psd_tools.api.adjustments.PatternFill attribute), 20
 - left (psd_tools.api.adjustments.SolidColorFill attribute), 17
 - left (psd_tools.api.layers.PixelLayer attribute), 40
 - left (psd_tools.api.layers.ShapeLayer attribute), 42
 - left (psd_tools.api.layers.SmartObjectLayer attribute), 45
 - left (psd_tools.api.layers.TypeLayer attribute), 48
 - left (psd_tools.api.mask.Mask attribute), 50
 - left (psd_tools.psd.layer_and_mask.LayerRecord attribute), 85, 86
 - left (psd_tools.psd.layer_and_mask.MaskData attribute), 88, 89
 - left (psd_tools.PSDImage attribute), 13
 - length (psd_tools.api.adjustments.GradientMap attribute), 27
 - length (psd_tools.psd.layer_and_mask.ChannelInfo attribute), 91
 - Levels (class in psd_tools.api.adjustments), 25
 - LEVELS (psd_tools.constants.TaggedBlockID attribute), 65
 - LIGHTEN (psd_tools.constants.BlendMode attribute), 57
 - LIGHTER_COLOR (psd_tools.constants.BlendMode attribute), 57
 - LIGHTROOM_WORKFLOW
 - line (class in psd_tools.api.shape), 53
 - line_alignment (psd_tools.api.shape.Stroke attribute), 52
 - line_cap_type (psd_tools.api.shape.Stroke attribute), 52
 - line_dash_offset (psd_tools.api.shape.Stroke attribute), 52
 - line_dash_set (psd_tools.api.shape.Stroke attribute), 52
 - line_end (psd_tools.api.shape.Line attribute), 53
 - line_join_type (psd_tools.api.shape.Stroke attribute), 52
 - line_start (psd_tools.api.shape.Line attribute), 53
 - line_weight (psd_tools.api.shape.Line attribute), 53
 - line_width (psd_tools.api.shape.Stroke attribute), 52
 - LINEAR_BURN (psd_tools.constants.BlendMode attribute), 57
 - LINEAR_DODGE (psd_tools.constants.BlendMode attribute), 57
 - LINEAR_LIGHT (psd_tools.constants.BlendMode attribute), 57
 - LINKED_LAYER1 (psd_tools.constants.TaggedBlockID attribute), 65
 - LINKED_LAYER2 (psd_tools.constants.TaggedBlockID attribute), 65
 - LINKED_LAYER3 (psd_tools.constants.TaggedBlockID attribute), 65
 - LINKED_LAYER_EXTERNAL (psd_tools.constants.TaggedBlockID attribute), 65
 - LinkedLayerType (class in psd_tools.constants), 59
 - List (class in psd_tools.psd.descriptor), 72
 - List (class in psd_tools.psd.engine_data), 78
 - luminosity (psd_tools.api.adjustments.ColorBalance attribute), 26
 - luminosity (psd_tools.api.adjustments.PhotoFilter attribute), 26
 - LUMINOSITY (psd_tools.constants.BlendMode attribute), 57
- ## M
- MAC_NSPRINTINFO (psd_tools.constants.ImageResourceID attribute), 61
 - MAC_PRINT_MANAGER_INFO (psd_tools.constants.ImageResourceID attribute), 61
 - magenta (psd_tools.api.adjustments.BlackAndWhite attribute), 26
 - Mask (class in psd_tools.api.mask), 50

- mask (psd_tools.api.adjustments.GradientFill attribute), 23
- mask (psd_tools.api.adjustments.PatternFill attribute), 20
- mask (psd_tools.api.adjustments.SolidColorFill attribute), 17
- mask (psd_tools.api.layers.Group attribute), 37
- mask (psd_tools.api.layers.PixelLayer attribute), 40
- mask (psd_tools.api.layers.ShapeLayer attribute), 42
- mask (psd_tools.api.layers.SmartObjectLayer attribute), 45
- mask (psd_tools.api.layers.TypeLayer attribute), 48
- mask_data (psd_tools.psd.layer_and_mask.LayerRecord attribute), 86
- mask_disabled (psd_tools.psd.layer_and_mask.MaskFlags attribute), 90
- MaskData (class in psd_tools.psd.layer_and_mask), 88
- MaskFlags (class in psd_tools.psd.layer_and_mask), 90
- MaskParameters (class in psd_tools.psd.layer_and_mask), 91
- master (psd_tools.api.adjustments.HueSaturation attribute), 25
- master (psd_tools.api.adjustments.Levels attribute), 25
- max_color (psd_tools.api.adjustments.GradientMap attribute), 27
- mean (psd_tools.api.adjustments.BrightnessContrast attribute), 24
- MEASUREMENT_SCALE (psd_tools.constants.ImageResourceID attribute), 61
- METADATA_SETTING (psd_tools.constants.TaggedBlockID attribute), 65
- method (psd_tools.api.adjustments.SelectiveColor attribute), 27
- midtone (psd_tools.api.adjustments.ColorBalance attribute), 26
- min_color (psd_tools.api.adjustments.GradientMap attribute), 27
- miter_limit (psd_tools.api.shape.Stroke attribute), 52
- mode (psd_tools.api.adjustments.GradientMap attribute), 27
- monochrome (psd_tools.api.adjustments.ChannelMixer attribute), 26
- MULTICHANNEL (psd_tools.constants.ColorMode attribute), 58
- MULTIPLY (psd_tools.constants.BlendMode attribute), 57
- N**
- Name (class in psd_tools.psd.descriptor), 73
- name (psd_tools.api.adjustments.GradientFill attribute), 23
- name (psd_tools.api.adjustments.PatternFill attribute), 20
- name (psd_tools.api.adjustments.SolidColorFill attribute), 17
- name (psd_tools.api.layers.Group attribute), 37
- name (psd_tools.api.layers.PixelLayer attribute), 40
- name (psd_tools.api.layers.ShapeLayer attribute), 42
- name (psd_tools.api.layers.SmartObjectLayer attribute), 45
- name (psd_tools.api.layers.TypeLayer attribute), 48
- name (psd_tools.psd.descriptor.Class attribute), 69
- name (psd_tools.psd.descriptor.Descriptor attribute), 70
- name (psd_tools.psd.descriptor.ObjectArray attribute), 73
- name (psd_tools.psd.descriptor.Property attribute), 73
- name (psd_tools.psd.image_resources.ImageResource attribute), 82
- name (psd_tools.psd.layer_and_mask.LayerRecord attribute), 86
- name (psd_tools.PSDImage attribute), 13
- NESTED_SECTION_DIVIDER_SETTING (psd_tools.constants.TaggedBlockID attribute), 65
- new() (psd_tools.psd.image_data.ImageData class method), 80
- new() (psd_tools.psd.image_resources.ImageResources class method), 82
- new() (psd_tools.PSDImage class method), 13
- noise (psd_tools.api.effects.DropShadow attribute), 28
- noise (psd_tools.api.effects.InnerGlow attribute), 31
- noise (psd_tools.api.effects.InnerShadow attribute), 29
- noise (psd_tools.api.effects.OuterGlow attribute), 30
- NON_BASE (psd_tools.constants.Clipping attribute), 58
- NORMAL (psd_tools.constants.BlendMode attribute), 57
- not_linked (psd_tools.psd.vector.VectorMaskSetting attribute), 95
- not_linked (psd_tools.api.shape.VectorMask attribute), 51
- O**
- OBJECT_BASED_EFFECTS_LAYER_INFO (psd_tools.constants.TaggedBlockID attribute), 66
- OBJECT_BASED_EFFECTS_LAYER_INFO_V0 (psd_tools.constants.TaggedBlockID attribute), 66
- OBJECT_BASED_EFFECTS_LAYER_INFO_V1 (psd_tools.constants.TaggedBlockID attribute), 66
- ObjectArray (class in psd_tools.psd.descriptor), 73
- obsolete (psd_tools.psd.layer_and_mask.LayerFlags attribute), 87
- OBSOLETE1 (psd_tools.constants.ImageResourceID attribute), 61
- OBSOLETE2 (psd_tools.constants.ImageResourceID attribute), 61
- OBSOLETE3 (psd_tools.constants.ImageResourceID attribute), 61

- OBSOLETE4 (psd_tools.constants.ImageResourceID attribute), 61
- OBSOLETE5 (psd_tools.constants.ImageResourceID attribute), 61
- Offset (class in psd_tools.psd.descriptor), 74
- offset (psd_tools.api.adjustments.Exposure attribute), 25
- offset (psd_tools.api.adjustments.GradientFill attribute), 23
- offset (psd_tools.api.adjustments.PatternFill attribute), 20
- offset (psd_tools.api.adjustments.SolidColorFill attribute), 17
- offset (psd_tools.api.effects.GradientOverlay attribute), 32
- offset (psd_tools.api.layers.Group attribute), 37
- offset (psd_tools.api.layers.PixelLayer attribute), 40
- offset (psd_tools.api.layers.ShapeLayer attribute), 42
- offset (psd_tools.api.layers.SmartObjectLayer attribute), 45
- offset (psd_tools.api.layers.TypeLayer attribute), 49
- offset (psd_tools.PSDImage attribute), 13
- ONION_SKINS (psd_tools.constants.ImageResourceID attribute), 61
- opacity (psd_tools.api.adjustments.GradientFill attribute), 23
- opacity (psd_tools.api.adjustments.PatternFill attribute), 20
- opacity (psd_tools.api.adjustments.SolidColorFill attribute), 18
- opacity (psd_tools.api.effects.BevelEmboss attribute), 34
- opacity (psd_tools.api.effects.ColorOverlay attribute), 31
- opacity (psd_tools.api.effects.DropShadow attribute), 28
- opacity (psd_tools.api.effects.GradientOverlay attribute), 32
- opacity (psd_tools.api.effects.InnerGlow attribute), 31
- opacity (psd_tools.api.effects.InnerShadow attribute), 29
- opacity (psd_tools.api.effects.OuterGlow attribute), 30
- opacity (psd_tools.api.effects.PatternOverlay attribute), 32
- opacity (psd_tools.api.effects.Satin attribute), 35
- opacity (psd_tools.api.effects.Stroke attribute), 33
- opacity (psd_tools.api.layers.Group attribute), 37
- opacity (psd_tools.api.layers.PixelLayer attribute), 40
- opacity (psd_tools.api.layers.ShapeLayer attribute), 43
- opacity (psd_tools.api.layers.SmartObjectLayer attribute), 45
- opacity (psd_tools.api.layers.TypeLayer attribute), 49
- opacity (psd_tools.api.shape.Stroke attribute), 52
- opacity (psd_tools.psd.layer_and_mask.GlobalLayerMaskInfo attribute), 84
- opacity (psd_tools.psd.layer_and_mask.LayerRecord attribute), 85, 86
- open() (psd_tools.api.smart_object.SmartObject method), 56
- open() (psd_tools.PSDImage class method), 14
- OPEN_FOLDER (psd_tools.constants.SectionDivider attribute), 64
- OPEN_KNOT_LINKED (psd_tools.constants.PathResourceID attribute), 63
- OPEN_KNOT_UNLINKED (psd_tools.constants.PathResourceID attribute), 63
- OPEN_LENGTH (psd_tools.constants.PathResourceID attribute), 63
- operation (psd_tools.psd.vector.Subpath attribute), 94
- ORIGIN_PATH_INFO (psd_tools.constants.ImageResourceID attribute), 61
- origin_type (psd_tools.api.shape.Ellipse attribute), 54
- origin_type (psd_tools.api.shape.Line attribute), 54
- origin_type (psd_tools.api.shape.Rectangle attribute), 55
- origin_type (psd_tools.api.shape.RoundedRectangle attribute), 55
- origination (psd_tools.api.adjustments.GradientFill attribute), 23
- origination (psd_tools.api.adjustments.PatternFill attribute), 20
- origination (psd_tools.api.adjustments.SolidColorFill attribute), 18
- origination (psd_tools.api.layers.Group attribute), 37
- origination (psd_tools.api.layers.PixelLayer attribute), 40
- origination (psd_tools.api.layers.ShapeLayer attribute), 43
- origination (psd_tools.api.layers.SmartObjectLayer attribute), 45
- origination (psd_tools.api.layers.TypeLayer attribute), 49
- ostype (psd_tools.psd.descriptor.Alias attribute), 69
- ostype (psd_tools.psd.descriptor.Bool attribute), 69
- ostype (psd_tools.psd.descriptor.Class1 attribute), 69
- ostype (psd_tools.psd.descriptor.Class2 attribute), 70
- ostype (psd_tools.psd.descriptor.Class3 attribute), 70
- ostype (psd_tools.psd.descriptor.Descriptor attribute), 70
- ostype (psd_tools.psd.descriptor.Double attribute), 70
- ostype (psd_tools.psd.descriptor.Enum attribute), 71
- ostype (psd_tools.psd.descriptor.EnumeratedReference attribute), 71
- ostype (psd_tools.psd.descriptor.GlobalObject attribute), 71
- ostype (psd_tools.psd.descriptor.Identifier attribute), 71
- ostype (psd_tools.psd.descriptor.Index attribute), 72
- ostype (psd_tools.psd.descriptor.Integer attribute), 72
- ostype (psd_tools.psd.descriptor.LargeInteger attribute), 72
- ostype (psd_tools.psd.descriptor.List attribute), 72
- ostype (psd_tools.psd.descriptor.Name attribute), 73
- ostype (psd_tools.psd.descriptor.ObjectArray attribute), 73
- ostype (psd_tools.psd.descriptor.Offset attribute), 74
- ostype (psd_tools.psd.descriptor.Path attribute), 74

- ostype (psd_tools.psd.descriptor.Property attribute), 73
- ostype (psd_tools.psd.descriptor.RawData attribute), 74
- ostype (psd_tools.psd.descriptor.Reference attribute), 75
- ostype (psd_tools.psd.descriptor.String attribute), 75
- ostype (psd_tools.psd.descriptor.UnitFloat attribute), 75
- ostype (psd_tools.psd.descriptor.UnitFloats attribute), 75
- OTHER (psd_tools.constants.SectionDivider attribute), 64
- OUTER_GLOW (psd_tools.constants.EffectOSType attribute), 59
- OuterGlow (class in psd_tools.api.effects), 29
- OVERLAY (psd_tools.constants.BlendMode attribute), 57
- overlay_color (psd_tools.psd.layer_and_mask.GlobalLayerMaskInfo attribute), 84
- overprint (psd_tools.api.effects.Stroke attribute), 33
- ## P
- PACK_BITS (psd_tools.constants.Compression attribute), 59
- parameters (psd_tools.api.mask.Mask attribute), 50
- parameters (psd_tools.psd.layer_and_mask.MaskData attribute), 88, 89
- parameters_applied (psd_tools.psd.layer_and_mask.MaskFlags attribute), 90
- parent (psd_tools.api.adjustments.GradientFill attribute), 23
- parent (psd_tools.api.adjustments.PatternFill attribute), 20
- parent (psd_tools.api.adjustments.SolidColorFill attribute), 18
- parent (psd_tools.api.layers.Group attribute), 37
- parent (psd_tools.api.layers.PixelLayer attribute), 40
- parent (psd_tools.api.layers.ShapeLayer attribute), 43
- parent (psd_tools.api.layers.SmartObjectLayer attribute), 46
- parent (psd_tools.api.layers.TypeLayer attribute), 49
- parent (psd_tools.PSDImage attribute), 14
- PASS_THROUGH (psd_tools.constants.BlendMode attribute), 57
- Path (class in psd_tools.psd.descriptor), 74
- path (psd_tools.psd.vector.VectorMaskSetting attribute), 95
- PATH_FILL (psd_tools.constants.PathResourceID attribute), 63
- PATH_INFO_0 (psd_tools.constants.ImageResourceID attribute), 61
- PATH_INFO_1 (psd_tools.constants.ImageResourceID attribute), 61
- PATH_INFO_2 (psd_tools.constants.ImageResourceID attribute), 61
- PATH_INFO_3 (psd_tools.constants.ImageResourceID attribute), 61
- PATH_INFO_4 (psd_tools.constants.ImageResourceID attribute), 61
- PATH_INFO_5 (psd_tools.constants.ImageResourceID attribute), 61
- PATH_INFO_6 (psd_tools.constants.ImageResourceID attribute), 61
- PATH_INFO_7 (psd_tools.constants.ImageResourceID attribute), 62
- PATH_INFO_8 (psd_tools.constants.ImageResourceID attribute), 62
- PATH_INFO_9 (psd_tools.constants.ImageResourceID attribute), 62
- PATH_INFO_990 (psd_tools.constants.ImageResourceID attribute), 62
- PATH_INFO_991 (psd_tools.constants.ImageResourceID attribute), 62
- PATH_INFO_992 (psd_tools.constants.ImageResourceID attribute), 62
- PATH_INFO_993 (psd_tools.constants.ImageResourceID attribute), 62
- PATH_INFO_994 (psd_tools.constants.ImageResourceID attribute), 62
- PATH_INFO_995 (psd_tools.constants.ImageResourceID attribute), 62
- PATH_INFO_996 (psd_tools.constants.ImageResourceID attribute), 62
- PATH_INFO_997 (psd_tools.constants.ImageResourceID attribute), 62
- PATH_SELECTION_STATE (psd_tools.constants.ImageResourceID attribute), 62
- PathResourceID (class in psd_tools.constants), 63
- paths (psd_tools.api.shape.VectorMask attribute), 51
- pattern (psd_tools.api.effects.PatternOverlay attribute), 32
- pattern (psd_tools.api.effects.Stroke attribute), 33
- PATTERN_DATA (psd_tools.constants.TaggedBlockID attribute), 66
- PATTERN_FILL_SETTING (psd_tools.constants.TaggedBlockID attribute), 66
- PatternFill (class in psd_tools.api.adjustments), 18
- PatternOverlay (class in psd_tools.api.effects), 32
- PATTERNS1 (psd_tools.constants.TaggedBlockID attribute), 66
- PATTERNS2 (psd_tools.constants.TaggedBlockID attribute), 66
- PATTERNS3 (psd_tools.constants.TaggedBlockID attribute), 66
- PER_LAYER (psd_tools.constants.GlobalLayerMaskKind attribute), 59
- phase (psd_tools.api.effects.PatternOverlay attribute), 32
- PHOTO_FILTER (psd_tools.constants.TaggedBlockID attribute), 66

PhotoFilter (class in psd_tools.api.adjustments), 26

photoshop_v5_later (psd_tools.psd.layer_and_mask.LayerFlags attribute), 87

PIN_LIGHT (psd_tools.constants.BlendMode attribute), 57

PIXEL_ASPECT_RATIO (psd_tools.constants.ImageResourceID attribute), 62

pixel_data_irrelevant (psd_tools.psd.layer_and_mask.LayerFlags attribute), 87

PIXEL_SOURCE_DATA1 (psd_tools.constants.TaggedBlockID attribute), 66

PIXEL_SOURCE_DATA2 (psd_tools.constants.TaggedBlockID attribute), 66

PixelLayer (class in psd_tools.api.layers), 38

PLACED_LAYER1 (psd_tools.constants.TaggedBlockID attribute), 66

PLACED_LAYER2 (psd_tools.constants.TaggedBlockID attribute), 66

PlacedLayerType (class in psd_tools.constants), 64

PLUGIN_RESOURCE_0 (psd_tools.constants.ImageResourceID attribute), 62

PLUGIN_RESOURCE_1 (psd_tools.constants.ImageResourceID attribute), 62

PLUGIN_RESOURCE_2 (psd_tools.constants.ImageResourceID attribute), 62

PLUGIN_RESOURCE_3 (psd_tools.constants.ImageResourceID attribute), 62

PLUGIN_RESOURCE_4 (psd_tools.constants.ImageResourceID attribute), 62

PLUGIN_RESOURCE_4990 (psd_tools.constants.ImageResourceID attribute), 62

PLUGIN_RESOURCE_4991 (psd_tools.constants.ImageResourceID attribute), 62

PLUGIN_RESOURCE_4992 (psd_tools.constants.ImageResourceID attribute), 62

PLUGIN_RESOURCE_4993 (psd_tools.constants.ImageResourceID attribute), 62

PLUGIN_RESOURCE_4994 (psd_tools.constants.ImageResourceID attribute), 62

PLUGIN_RESOURCE_4995 (psd_tools.constants.ImageResourceID attribute), 62

PLUGIN_RESOURCE_4996 (psd_tools.constants.ImageResourceID attribute), 62

PLUGIN_RESOURCE_4997 (psd_tools.constants.ImageResourceID attribute), 62

PLUGIN_RESOURCE_4998 (psd_tools.constants.ImageResourceID attribute), 62

PLUGIN_RESOURCE_4999 (psd_tools.constants.ImageResourceID attribute), 62

PLUGIN_RESOURCE_5 (psd_tools.constants.ImageResourceID attribute), 62

PLUGIN_RESOURCE_6 (psd_tools.constants.ImageResourceID attribute), 62

PLUGIN_RESOURCE_7 (psd_tools.constants.ImageResourceID attribute), 62

PLUGIN_RESOURCE_8 (psd_tools.constants.ImageResourceID attribute), 62

PLUGIN_RESOURCE_9 (psd_tools.constants.ImageResourceID attribute), 62

pos_relative_to_layer (psd_tools.psd.layer_and_mask.MaskFlags attribute), 90

position (psd_tools.api.effects.Stroke attribute), 33

Posterize (class in psd_tools.api.adjustments), 27

posterize (psd_tools.api.adjustments.Posterize attribute), 27

POSTERIZE (psd_tools.constants.TaggedBlockID attribute), 66

preceding (psd_tools.psd.vector.Knot attribute), 95

present (psd_tools.api.effects.BevelEmboss attribute), 34

present (psd_tools.api.effects.ColorOverlay attribute), 31

present (psd_tools.api.effects.DropShadow attribute), 28

present (psd_tools.api.effects.GradientOverlay attribute), 32

present (psd_tools.api.effects.InnerGlow attribute), 31

present (psd_tools.api.effects.InnerShadow attribute), 29

present (psd_tools.api.effects.OuterGlow attribute), 30

present (psd_tools.api.effects.PatternOverlay attribute), 32

present (psd_tools.api.effects.Satin attribute), 35

present (psd_tools.api.effects.Stroke attribute), 33

preset_file_name (psd_tools.api.adjustments.BlackAndWhite attribute), 26

preset_kind (psd_tools.api.adjustments.BlackAndWhite attribute), 26

PRINT_FLAGS (psd_tools.constants.ImageResourceID attribute), 62

- attribute), 62
- PRINT_FLAGS_INFO (psd_tools.constants.ImageResourceID attribute), 62
- PRINT_INFO_CS2 (psd_tools.constants.ImageResourceID attribute), 62
- PRINT_INFO_CS5 (psd_tools.constants.ImageResourceID attribute), 63
- PRINT_SCALE (psd_tools.constants.ImageResourceID attribute), 63
- PRINT_STYLE (psd_tools.constants.ImageResourceID attribute), 63
- PrintScaleStyle (class in psd_tools.constants), 64
- Property (class in psd_tools.psd.descriptor), 73
- Property (class in psd_tools.psd.engine_data), 78
- PROTECTED_SETTING (psd_tools.constants.TaggedBlockID attribute), 66
- PSD (class in psd_tools.psd), 67
- psd_tools.api.adjustments (module), 15
- psd_tools.api.effects (module), 27
- psd_tools.api.layers (module), 35
- psd_tools.api.mask (module), 50
- psd_tools.api.shape (module), 51
- psd_tools.api.smart_object (module), 56
- psd_tools.constants (module), 56
- psd_tools.psd (module), 67
- psd_tools.psd.color_mode_data (module), 68
- psd_tools.psd.descriptor (module), 68
- psd_tools.psd.engine_data (module), 76
- psd_tools.psd.header (module), 79
- psd_tools.psd.image_data (module), 80
- psd_tools.psd.image_resources (module), 81
- psd_tools.psd.layer_and_mask (module), 83
- psd_tools.psd.tagged_blocks (module), 93
- psd_tools.psd.vector (module), 94
- PSDImage (class in psd_tools), 11
- ## Q
- quality_jitter (psd_tools.api.effects.InnerGlow attribute), 31
- quality_jitter (psd_tools.api.effects.OuterGlow attribute), 30
- quality_range (psd_tools.api.effects.InnerGlow attribute), 31
- quality_range (psd_tools.api.effects.OuterGlow attribute), 30
- QUICK_MASK_INFO (psd_tools.constants.ImageResourceID attribute), 63
- ## R
- radii (psd_tools.api.shape.RoundedRectangle attribute), 55
- random_seed (psd_tools.api.adjustments.GradientMap attribute), 27
- RASTER (psd_tools.constants.PlacedLayerType attribute), 64
- RAW (psd_tools.constants.Compression attribute), 59
- RawData (class in psd_tools.psd.descriptor), 74
- read() (psd_tools.psd.color_mode_data.ColorModeData class method), 68
- read() (psd_tools.psd.descriptor.Bool class method), 69
- read() (psd_tools.psd.descriptor.Class class method), 69
- read() (psd_tools.psd.descriptor.Descriptor class method), 70
- read() (psd_tools.psd.descriptor.Double class method), 70
- read() (psd_tools.psd.descriptor.Enum class method), 71
- read() (psd_tools.psd.descriptor.EnumeratedReference class method), 71
- read() (psd_tools.psd.descriptor.Integer class method), 72
- read() (psd_tools.psd.descriptor.LargeInteger class method), 72
- read() (psd_tools.psd.descriptor.List class method), 72
- read() (psd_tools.psd.descriptor.Name class method), 73
- read() (psd_tools.psd.descriptor.ObjectArray class method), 73
- read() (psd_tools.psd.descriptor.Offset class method), 74
- read() (psd_tools.psd.descriptor.Property class method), 73
- read() (psd_tools.psd.descriptor.RawData class method), 74
- read() (psd_tools.psd.descriptor.UnitFloat class method), 75
- read() (psd_tools.psd.descriptor.UnitFloats class method), 75
- read() (psd_tools.psd.engine_data.Bool class method), 77
- read() (psd_tools.psd.engine_data.Dict class method), 77
- read() (psd_tools.psd.engine_data.Float class method), 77
- read() (psd_tools.psd.engine_data.Integer class method), 77
- read() (psd_tools.psd.engine_data.List class method), 78
- read() (psd_tools.psd.engine_data.Property class method), 78
- read() (psd_tools.psd.engine_data.String class method), 78
- read() (psd_tools.psd.header.FileHeader class method), 79
- read() (psd_tools.psd.image_data.ImageData class method), 80
- read() (psd_tools.psd.image_resources.ImageResource class method), 82
- read() (psd_tools.psd.image_resources.ImageResources class method), 82
- read() (psd_tools.psd.layer_and_mask.ChannelData class method), 93
- read() (psd_tools.psd.layer_and_mask.ChannelDataList class method), 92
- read() (psd_tools.psd.layer_and_mask.ChannelsImageData class method), 92

read() (psd_tools.psd.layer_and_mask.ChannelInfo class method), 91

read() (psd_tools.psd.layer_and_mask.GlobalLayerMaskInfo class method), 85

read() (psd_tools.psd.layer_and_mask.LayerAndMaskInfo class method), 83

read() (psd_tools.psd.layer_and_mask.LayerBlendingRanges class method), 88

read() (psd_tools.psd.layer_and_mask.LayerFlags class method), 87

read() (psd_tools.psd.layer_and_mask.LayerInfo class method), 84

read() (psd_tools.psd.layer_and_mask.LayerRecord class method), 86

read() (psd_tools.psd.layer_and_mask.LayerRecords class method), 85

read() (psd_tools.psd.layer_and_mask.MaskData class method), 89

read() (psd_tools.psd.layer_and_mask.MaskFlags class method), 90

read() (psd_tools.psd.layer_and_mask.MaskParameters class method), 91

read() (psd_tools.psd.PSD class method), 68

read() (psd_tools.psd.vector.Knot class method), 95

read() (psd_tools.psd.vector.Subpath class method), 94

real_background_color (psd_tools.psd.layer_and_mask.MaskData attribute), 89

real_bottom (psd_tools.psd.layer_and_mask.MaskData attribute), 89

real_flags (psd_tools.api.mask.Mask attribute), 50

real_flags (psd_tools.psd.layer_and_mask.MaskData attribute), 89

real_height (psd_tools.psd.layer_and_mask.MaskData attribute), 89

real_left (psd_tools.psd.layer_and_mask.MaskData attribute), 89

real_right (psd_tools.psd.layer_and_mask.MaskData attribute), 89

real_top (psd_tools.psd.layer_and_mask.MaskData attribute), 89

REAL_USER_LAYER_MASK (psd_tools.constants.ChannelID attribute), 58

real_width (psd_tools.psd.layer_and_mask.MaskData attribute), 89

Rectangle (class in psd_tools.api.shape), 54

red (psd_tools.api.adjustments.BlackAndWhite attribute), 26

Reference (class in psd_tools.psd.descriptor), 75

REFERENCE_POINT (psd_tools.constants.TaggedBlockID attribute), 66

resolution (psd_tools.api.shape.Ellipse attribute), 54

resolution (psd_tools.api.shape.Line attribute), 54

resolution (psd_tools.api.shape.Rectangle attribute), 55

resolution (psd_tools.api.shape.RoundedRectangle attribute), 55

resolution (psd_tools.api.smart_object.SmartObject attribute), 56

RESOLUTION_INFO (psd_tools.constants.ImageResourceID attribute), 63

resource_dict (psd_tools.api.layers.TypeLayer attribute), 49

reversed (psd_tools.api.adjustments.GradientMap attribute), 27

reversed (psd_tools.api.effects.GradientOverlay attribute), 32

RGB (psd_tools.constants.ColorMode attribute), 58

RGB (psd_tools.constants.ColorSpaceID attribute), 58

right (psd_tools.api.adjustments.GradientFill attribute), 23

right (psd_tools.api.adjustments.PatternFill attribute), 20

right (psd_tools.api.adjustments.SolidColorFill attribute), 18

right (psd_tools.api.layers.PixelLayer attribute), 40

right (psd_tools.api.layers.ShapeLayer attribute), 43

right (psd_tools.api.layers.SmartObjectLayer attribute), 46

right (psd_tools.api.layers.TypeLayer attribute), 49

right (psd_tools.api.mask.Mask attribute), 50

right (psd_tools.psd.layer_and_mask.LayerRecord attribute), 85, 86

right (psd_tools.psd.layer_and_mask.MaskData attribute), 88, 89

right (psd_tools.PSDImage attribute), 14

roughness (psd_tools.api.adjustments.GradientMap attribute), 27

RoundedRectangle (class in psd_tools.api.shape), 55

S

Satin (class in psd_tools.api.effects), 34

saturation (psd_tools.api.adjustments.Vibrance attribute), 25

SATURATION (psd_tools.constants.BlendMode attribute), 57

save() (psd_tools.api.smart_object.SmartObject method), 56

save() (psd_tools.PSDImage method), 14

SAVING_MERGED_TRANSPARENCY (psd_tools.constants.TaggedBlockID attribute), 66

SAVING_MERGED_TRANSPARENCY16 (psd_tools.constants.TaggedBlockID attribute), 66

SAVING_MERGED_TRANSPARENCY32 (psd_tools.constants.TaggedBlockID attribute), 66

scale (psd_tools.api.effects.Effects attribute), 28

scale (psd_tools.api.effects.GradientOverlay attribute), 32

- scale (psd_tools.api.effects.PatternOverlay attribute), 32
- SCREEN (psd_tools.constants.BlendMode attribute), 57
- SECTION_DIVIDER_SETTING (psd_tools.constants.TaggedBlockID attribute), 66
- SectionDivider (class in psd_tools.constants), 64
- SELECTIVE_COLOR (psd_tools.constants.TaggedBlockID attribute), 66
- SelectiveColor (class in psd_tools.api.adjustments), 27
- set_data() (psd_tools.psd.image_data.ImageData method), 80
- set_data() (psd_tools.psd.layer_and_mask.ChannelData method), 93
- shadow_color (psd_tools.api.effects.BevelEmboss attribute), 34
- shadow_mode (psd_tools.api.effects.BevelEmboss attribute), 34
- shadow_opacity (psd_tools.api.effects.BevelEmboss attribute), 34
- shadows (psd_tools.api.adjustments.ColorBalance attribute), 26
- ShapeLayer (class in psd_tools.api.layers), 41
- SHEET_COLOR_SETTING (psd_tools.constants.TaggedBlockID attribute), 66
- SHEET_DISCLOSURE (psd_tools.constants.ImageResourceID attribute), 63
- show_transparency (psd_tools.api.adjustments.GradientMap attribute), 27
- shown (psd_tools.api.effects.BevelEmboss attribute), 34
- shown (psd_tools.api.effects.ColorOverlay attribute), 31
- shown (psd_tools.api.effects.DropShadow attribute), 28
- shown (psd_tools.api.effects.GradientOverlay attribute), 32
- shown (psd_tools.api.effects.InnerGlow attribute), 31
- shown (psd_tools.api.effects.InnerShadow attribute), 29
- shown (psd_tools.api.effects.OuterGlow attribute), 30
- shown (psd_tools.api.effects.PatternOverlay attribute), 32
- shown (psd_tools.api.effects.Satin attribute), 35
- shown (psd_tools.api.effects.Stroke attribute), 33
- signature (psd_tools.psd.header.FileHeader attribute), 79
- signature (psd_tools.psd.image_resources.ImageResource attribute), 82
- signature (psd_tools.psd.layer_and_mask.LayerRecord attribute), 85, 86
- size (psd_tools.api.adjustments.GradientFill attribute), 23
- size (psd_tools.api.adjustments.PatternFill attribute), 21
- size (psd_tools.api.adjustments.SolidColorFill attribute), 18
- size (psd_tools.api.effects.BevelEmboss attribute), 34
- size (psd_tools.api.effects.DropShadow attribute), 28
- size (psd_tools.api.effects.InnerGlow attribute), 31
- size (psd_tools.api.effects.InnerShadow attribute), 29
- size (psd_tools.api.effects.OuterGlow attribute), 30
- size (psd_tools.api.effects.Satin attribute), 35
- size (psd_tools.api.effects.Stroke attribute), 33
- size (psd_tools.api.layers.Group attribute), 37
- size (psd_tools.api.layers.PixelLayer attribute), 40
- size (psd_tools.api.layers.ShapeLayer attribute), 43
- size (psd_tools.api.layers.SmartObjectLayer attribute), 46
- size (psd_tools.api.layers.TypeLayer attribute), 49
- size (psd_tools.api.mask.Mask attribute), 50
- size (psd_tools.PSDImage attribute), 14
- SIZE_TO_FIT (psd_tools.constants.PrintScaleStyle attribute), 64
- SLICES (psd_tools.constants.ImageResourceID attribute), 63
- smart_object (psd_tools.api.layers.SmartObjectLayer attribute), 46
- SMART_OBJECT_LAYER_DATA1 (psd_tools.constants.TaggedBlockID attribute), 66
- SMART_OBJECT_LAYER_DATA2 (psd_tools.constants.TaggedBlockID attribute), 66
- SmartObject (class in psd_tools.api.smart_object), 56
- SmartObjectLayer (class in psd_tools.api.layers), 43
- SOFT_LIGHT (psd_tools.constants.BlendMode attribute), 57
- shown (psd_tools.api.effects.BevelEmboss attribute), 34
- SOLID_COLOR_SHEET_SETTING (psd_tools.constants.TaggedBlockID attribute), 66
- SOLID_FILL (psd_tools.constants.EffectOSType attribute), 59
- SolidColorFill (class in psd_tools.api.adjustments), 16
- SPOT_HALFTONE (psd_tools.constants.ImageResourceID attribute), 63
- String (class in psd_tools.psd.descriptor), 75
- String (class in psd_tools.psd.engine_data), 78
- Stroke (class in psd_tools.api.effects), 33
- Stroke (class in psd_tools.api.shape), 52
- stroke (psd_tools.api.layers.ShapeLayer attribute), 43
- stroke_adjust (psd_tools.api.shape.Stroke attribute), 52
- Subpath (class in psd_tools.psd.vector), 94
- SUBTRACT (psd_tools.constants.BlendMode attribute), 57
- ## T
- tagged_blocks (psd_tools.psd.layer_and_mask.LayerAndMaskInformation attribute), 83
- tagged_blocks (psd_tools.psd.layer_and_mask.LayerRecord attribute), 86
- tagged_blocks (psd_tools.PSDImage attribute), 14
- TaggedBlock (class in psd_tools.psd.tagged_blocks), 94
- TaggedBlockID (class in psd_tools.constants), 64
- TaggedBlocks (class in psd_tools.psd.tagged_blocks), 93
- text (psd_tools.api.layers.TypeLayer attribute), 49

- TEXT_ENGINE_DATA (psd_tools.constants.TaggedBlockID attribute), 66
 - Threshold (class in psd_tools.api.adjustments), 27
 - threshold (psd_tools.api.adjustments.Threshold attribute), 27
 - THRESHOLD (psd_tools.constants.TaggedBlockID attribute), 66
 - thumbnail() (psd_tools.PSDImage method), 14
 - THUMBNAIL_RESOURCE (psd_tools.constants.ImageResourceID attribute), 63
 - THUMBNAIL_RESOURCE_PS4 (psd_tools.constants.ImageResourceID attribute), 63
 - TIMELINE_INFO (psd_tools.constants.ImageResourceID attribute), 63
 - tint_color (psd_tools.api.adjustments.BlackAndWhite attribute), 26
 - tobytes() (psd_tools.psd.PSD method), 68
 - top (psd_tools.api.adjustments.GradientFill attribute), 23
 - top (psd_tools.api.adjustments.PatternFill attribute), 21
 - top (psd_tools.api.adjustments.SolidColorFill attribute), 18
 - top (psd_tools.api.layers.PixelLayer attribute), 40
 - top (psd_tools.api.layers.ShapeLayer attribute), 43
 - top (psd_tools.api.layers.SmartObjectLayer attribute), 46
 - top (psd_tools.api.layers.TypeLayer attribute), 49
 - top (psd_tools.api.mask.Mask attribute), 50
 - top (psd_tools.psd.layer_and_mask.LayerRecord attribute), 85, 86
 - top (psd_tools.psd.layer_and_mask.MaskData attribute), 88, 89
 - top (psd_tools.PSDImage attribute), 14
 - topil() (psd_tools.api.adjustments.GradientFill method), 23
 - topil() (psd_tools.api.adjustments.PatternFill method), 21
 - topil() (psd_tools.api.adjustments.SolidColorFill method), 18
 - topil() (psd_tools.api.layers.Group method), 38
 - topil() (psd_tools.api.layers.PixelLayer method), 40
 - topil() (psd_tools.api.layers.ShapeLayer method), 43
 - topil() (psd_tools.api.layers.SmartObjectLayer method), 46
 - topil() (psd_tools.api.layers.TypeLayer method), 49
 - topil() (psd_tools.api.mask.Mask method), 51
 - topil() (psd_tools.PSDImage method), 14
 - transform (psd_tools.api.layers.TypeLayer attribute), 49
 - TRANSPARENCY_INDEX (psd_tools.constants.ImageResourceID attribute), 63
 - TRANSPARENCY_MASK (psd_tools.constants.ChannelID attribute), 58
 - transparency_protected (psd_tools.psd.layer_and_mask.LayerFlags attribute), 87
 - TRANSPARENCY_SHAPES_LAYER (psd_tools.constants.TaggedBlockID attribute), 66
 - transparency_stops (psd_tools.api.adjustments.GradientMap attribute), 27
 - type (psd_tools.api.effects.GradientOverlay attribute), 32
 - TYPE_TOOL_INFO (psd_tools.constants.TaggedBlockID attribute), 66
 - TYPE_TOOL_OBJECT_SETTING (psd_tools.constants.TaggedBlockID attribute), 66
 - TypeLayer (class in psd_tools.api.layers), 46
- ## U
- undocumented_1 (psd_tools.psd.layer_and_mask.LayerFlags attribute), 87
 - undocumented_1 (psd_tools.psd.layer_and_mask.MaskFlags attribute), 90
 - undocumented_2 (psd_tools.psd.layer_and_mask.LayerFlags attribute), 87
 - undocumented_2 (psd_tools.psd.layer_and_mask.MaskFlags attribute), 90
 - undocumented_3 (psd_tools.psd.layer_and_mask.LayerFlags attribute), 87
 - undocumented_3 (psd_tools.psd.layer_and_mask.MaskFlags attribute), 90
 - UNICODE_LAYER_NAME (psd_tools.constants.TaggedBlockID attribute), 66
 - UNICODE_PATH_NAME (psd_tools.constants.TaggedBlockID attribute), 66
 - unique_id (psd_tools.api.smart_object.SmartObject attribute), 56
 - unit (psd_tools.psd.descriptor.UnitFloat attribute), 75
 - unit (psd_tools.psd.descriptor.UnitFloats attribute), 75
 - UnitFloat (class in psd_tools.psd.descriptor), 75
 - UnitFloats (class in psd_tools.psd.descriptor), 75
 - UNKNOWN (psd_tools.constants.PlacedLayerType attribute), 64
 - URL (psd_tools.constants.ImageResourceID attribute), 63
 - URL_LIST (psd_tools.constants.ImageResourceID attribute), 63
 - use_global_light (psd_tools.api.effects.BevelEmboss attribute), 34
 - use_global_light (psd_tools.api.effects.DropShadow attribute), 28
 - use_global_light (psd_tools.api.effects.InnerShadow attribute), 29
 - use_legacy (psd_tools.api.adjustments.BrightnessContrast attribute), 24

- use_shape (psd_tools.api.effects.BevelEmboss attribute), 34
- use_texture (psd_tools.api.effects.BevelEmboss attribute), 34
- use_tint (psd_tools.api.adjustments.BlackAndWhite attribute), 26
- use_vector_color (psd_tools.api.adjustments.GradientMap attribute), 27
- USER_DEFINED (psd_tools.constants.PrintScaleStyle attribute), 64
- USER_LAYER_MASK (psd_tools.constants.ChannelID attribute), 58
- USER_MASK (psd_tools.constants.TaggedBlockID attribute), 66
- user_mask_density (psd_tools.psd.layer_and_mask.MaskParameters attribute), 91
- user_mask_feather (psd_tools.psd.layer_and_mask.MaskParameters attribute), 91
- user_mask_from_render (psd_tools.psd.layer_and_mask.MaskFlags attribute), 90
- USING_ALIGNED_RENDERING (psd_tools.constants.TaggedBlockID attribute), 66
- ## V
- validate() (psd_tools.psd.PSD method), 68
- value (psd_tools.psd.color_mode_data.ColorModeData attribute), 68
- value (psd_tools.psd.descriptor.Alias attribute), 69
- value (psd_tools.psd.descriptor.Bool attribute), 69
- value (psd_tools.psd.descriptor.Double attribute), 70
- value (psd_tools.psd.descriptor.Enum attribute), 71
- value (psd_tools.psd.descriptor.EnumeratedReference attribute), 71
- value (psd_tools.psd.descriptor.Integer attribute), 72
- value (psd_tools.psd.descriptor.LargeInteger attribute), 72
- value (psd_tools.psd.descriptor.Offset attribute), 74
- value (psd_tools.psd.descriptor.RawData attribute), 74
- value (psd_tools.psd.descriptor.String attribute), 75
- value (psd_tools.psd.descriptor.UnitFloat attribute), 75
- values (psd_tools.psd.descriptor.UnitFloats attribute), 75
- VECTOR (psd_tools.constants.PlacedLayerType attribute), 64
- vector_mask (psd_tools.api.adjustments.GradientFill attribute), 24
- vector_mask (psd_tools.api.adjustments.PatternFill attribute), 21
- vector_mask (psd_tools.api.adjustments.SolidColorFill attribute), 18
- vector_mask (psd_tools.api.layers.Group attribute), 38
- vector_mask (psd_tools.api.layers.PixelLayer attribute), 40
- vector_mask (psd_tools.api.layers.ShapeLayer attribute), 43
- vector_mask (psd_tools.api.layers.SmartObjectLayer attribute), 46
- vector_mask (psd_tools.api.layers.TypeLayer attribute), 49
- VECTOR_MASK_AS_GLOBAL_MASK (psd_tools.constants.TaggedBlockID attribute), 66
- vector_mask_density (psd_tools.psd.layer_and_mask.MaskParameters attribute), 91
- vector_mask_feather (psd_tools.psd.layer_and_mask.MaskParameters attribute), 91
- VECTOR_MASK_SETTING1 (psd_tools.constants.TaggedBlockID attribute), 66
- VECTOR_MASK_SETTING2 (psd_tools.constants.TaggedBlockID attribute), 67
- VECTOR_ORIGINATION_DATA (psd_tools.constants.TaggedBlockID attribute), 67
- VECTOR_STROKE_CONTENT_DATA (psd_tools.constants.TaggedBlockID attribute), 67
- VECTOR_STROKE_DATA (psd_tools.constants.TaggedBlockID attribute), 67
- VectorMask (class in psd_tools.api.shape), 51
- VectorMaskSetting (class in psd_tools.psd.vector), 95
- VectorStrokeContentSetting (class in psd_tools.psd.vector), 95
- version (psd_tools.psd.header.FileHeader attribute), 79
- version (psd_tools.psd.vector.VectorMaskSetting attribute), 95
- version (psd_tools.psd.vector.VectorStrokeContentSetting attribute), 95
- version (psd_tools.PSDImage attribute), 14
- VERSION_INFO (psd_tools.constants.ImageResourceID attribute), 63
- Vibrance (class in psd_tools.api.adjustments), 25
- vibrance (psd_tools.api.adjustments.Vibrance attribute), 25
- VIBRANCE (psd_tools.constants.TaggedBlockID attribute), 67
- viewbox (psd_tools.PSDImage attribute), 14
- visible (psd_tools.api.adjustments.GradientFill attribute), 24
- visible (psd_tools.api.adjustments.PatternFill attribute), 21
- visible (psd_tools.api.adjustments.SolidColorFill attribute), 18
- visible (psd_tools.api.layers.Group attribute), 38
- visible (psd_tools.api.layers.PixelLayer attribute), 41

- visible (psd_tools.api.layers.ShapeLayer attribute), 43
 - visible (psd_tools.api.layers.SmartObjectLayer attribute), 46
 - visible (psd_tools.api.layers.TypeLayer attribute), 49
 - visible (psd_tools.psd.layer_and_mask.LayerFlags attribute), 87
 - visible (psd_tools.PSDImage attribute), 14
 - VIVID_LIGHT (psd_tools.constants.BlendMode attribute), 57
 - vrsn (psd_tools.api.adjustments.BrightnessContrast attribute), 24
- ## W
- warp (psd_tools.api.layers.TypeLayer attribute), 50
 - warp (psd_tools.api.smart_object.SmartObject attribute), 56
 - WATERMARK (psd_tools.constants.ImageResourceID attribute), 63
 - width (psd_tools.api.adjustments.GradientFill attribute), 24
 - width (psd_tools.api.adjustments.PatternFill attribute), 21
 - width (psd_tools.api.adjustments.SolidColorFill attribute), 18
 - width (psd_tools.api.layers.Group attribute), 38
 - width (psd_tools.api.layers.PixelLayer attribute), 41
 - width (psd_tools.api.layers.ShapeLayer attribute), 43
 - width (psd_tools.api.layers.SmartObjectLayer attribute), 46
 - width (psd_tools.api.layers.TypeLayer attribute), 50
 - width (psd_tools.api.mask.Mask attribute), 51
 - width (psd_tools.psd.header.FileHeader attribute), 79, 80
 - width (psd_tools.psd.layer_and_mask.LayerRecord attribute), 86
 - width (psd_tools.psd.layer_and_mask.MaskData attribute), 89
 - width (psd_tools.PSDImage attribute), 14
 - WINDOWS_DEVMODE (psd_tools.constants.ImageResourceID attribute), 63
 - WORKFLOW_URL (psd_tools.constants.ImageResourceID attribute), 63
 - WORKING_PATH (psd_tools.constants.ImageResourceID attribute), 63
 - write() (psd_tools.psd.color_mode_data.ColorModeData method), 68
 - write() (psd_tools.psd.descriptor.Bool method), 69
 - write() (psd_tools.psd.descriptor.Class method), 69
 - write() (psd_tools.psd.descriptor.Descriptor method), 70
 - write() (psd_tools.psd.descriptor.Double method), 70
 - write() (psd_tools.psd.descriptor.Enum method), 71
 - write() (psd_tools.psd.descriptor.EnumeratedReference method), 71
 - write() (psd_tools.psd.descriptor.Integer method), 72
 - write() (psd_tools.psd.descriptor.LargeInteger method), 72
 - write() (psd_tools.psd.descriptor.List method), 72
 - write() (psd_tools.psd.descriptor.Name method), 73
 - write() (psd_tools.psd.descriptor.ObjectArray method), 73
 - write() (psd_tools.psd.descriptor.Offset method), 74
 - write() (psd_tools.psd.descriptor.Property method), 74
 - write() (psd_tools.psd.descriptor.RawData method), 74
 - write() (psd_tools.psd.descriptor.UnitFloat method), 75
 - write() (psd_tools.psd.descriptor.UnitFloats method), 75
 - write() (psd_tools.psd.engine_data.Bool method), 77
 - write() (psd_tools.psd.engine_data.Dict method), 77
 - write() (psd_tools.psd.engine_data.EngineData2 method), 76
 - write() (psd_tools.psd.engine_data.Float method), 77
 - write() (psd_tools.psd.engine_data.Integer method), 78
 - write() (psd_tools.psd.engine_data.List method), 78
 - write() (psd_tools.psd.engine_data.Property method), 78
 - write() (psd_tools.psd.engine_data.String method), 78
 - write() (psd_tools.psd.header.FileHeader method), 80
 - write() (psd_tools.psd.image_data.ImageData method), 81
 - write() (psd_tools.psd.image_resources.ImageResource method), 83
 - write() (psd_tools.psd.image_resources.ImageResources method), 82
 - write() (psd_tools.psd.layer_and_mask.ChannelData method), 93
 - write() (psd_tools.psd.layer_and_mask.ChannelImageData method), 92
 - write() (psd_tools.psd.layer_and_mask.ChannelInfo method), 91
 - write() (psd_tools.psd.layer_and_mask.GlobalLayerMaskInfo method), 85
 - write() (psd_tools.psd.layer_and_mask.LayerAndMaskInformation method), 83
 - write() (psd_tools.psd.layer_and_mask.LayerBlendingRanges method), 88
 - write() (psd_tools.psd.layer_and_mask.LayerFlags method), 87
 - write() (psd_tools.psd.layer_and_mask.LayerInfo method), 84
 - write() (psd_tools.psd.layer_and_mask.LayerRecord method), 87
 - write() (psd_tools.psd.layer_and_mask.MaskData method), 89
 - write() (psd_tools.psd.layer_and_mask.MaskFlags method), 90
 - write() (psd_tools.psd.layer_and_mask.MaskParameters method), 91
 - write() (psd_tools.psd.PSD method), 68
 - write() (psd_tools.psd.vector.Knot method), 95
 - write() (psd_tools.psd.vector.Subpath method), 94

X

XMP_METADATA (psd_tools.constants.ImageResourceID attribute), 63

xyz (psd_tools.api.adjustments.PhotoFilter attribute), 26

Y

yellow (psd_tools.api.adjustments.BlackAndWhite attribute), 26

Z

ZIP (psd_tools.constants.Compression attribute), 59

ZIP_WITH_PREDICTION
(psd_tools.constants.Compression attribute),
59