# Images

## Introduction

Image variations (image aliases) allow you to define and use different versions of the same image. You generate variations based on filters which modify aspects such as size and proportions, quality or decorations.

Image variations are generated with LiipImagineBundle, using the underlying Imagine library from avalanche123. This bundle supports GD, Imagick or Gmagick PHP extensions, and allows you to define flexible filters in PHP. Image files are stored using the `IOService`, and are completely independent from the `ezimage` Field Type. They are generated only once and cleared on demand (e.g. on content removal).

## Configuration

Custom image variations are defined in `ezplatform.yml` or any imported semantic configuration file. The definition is dynamic, so it can be configured per siteaccess and all the other scopes.

**Example image variation definition**

```
ezpublish:
    system:
        my_siteaccess:
            image_variations:
                small:
                    reference: null
                    filters:
                        - { name:
geometry/scaledownonly, params: [100, 160] }
                medium:
                    reference: null
                    filters:
                        - { name:
geometry/scaledownonly, params: [200, 290] }
                listitem:
                    reference: null
                    filters:
                        - { name:
geometry/scaledownonly, params: [130, 190] }
                articleimage:
                    reference: null
                    filters:
                        - { name: geometry/scalewidth,
params: [770] }
```

**Important**
Each variation name **must be unique**. It may contain _ or - or numbers, but no spaces.

The following parameters are set for each variation:

- `reference`: Name of a reference variation to base the variation on. If set to `null` (or ~, which means `null` in YAML), the variation will take the original image for reference. It can be any available variation configured in the `ezpublish` namespace, or a `filter_set` de fined in the `liip_imagine` namespace.
- `filters`: Array of filter definitions (hashes containing `name` and `params` keys). See possible values below.

**In this topic:**

### Default image variations

A few basic image variations are included by default in eZ Platform in the `default_settings.yml` config file:

```
ezsettings.default.image_variations:
    reference:
        reference: ~
        filters:
            geometry/scaledownonly: [600, 600]
    small:
        reference: reference
        filters:
            geometry/scaledownonly: [100, 100]
    tiny:
        reference: reference
        filters:
            geometry/scaledownonly: [30, 30]
    medium:
        reference: reference
        filters:
            geometry/scaledownonly: [200, 200]
    large:
        reference: reference
        filters:
            geometry/scaledownonly: [300, 300]
```

## Usage

### Filter usage examples

#### Scaling with an eZ Platform filter

This configuration defines a `medium` image variation that is scaled to a width of 700 px.

```
ezpublish:
    system:
        my_siteaccess:
            image_variations:
                medium:
                    reference: null
                    filters:
                        - geometry/scalewidth:
             params: [770]
```

#### Image quality with a liip filter

This configuration adds a limit to the image quality using a liip filter.

You can use both an eZ Platform and a liip filter for the same image variation, in this case `medium`.

```
ezpublish:
    system:
        my_siteaccess:
            image_variations:
    # List of variations

liip_imagine:
    driver: imagick
    filter_sets:
        medium:
            jpeg_quality: 50
```

> Notice that the `liip_imagine` key is not placed under `image_variations`, but at the same level as `ezpublish`.

## Post-Processors

LiipImagineBundle supports post-processors on image aliases. It is possible to specify them in image variation configuration:

```
ezpublish:
    system:
        my_siteaccess:
            image_variations:
                articleimage:
                    reference: null
                    filters:
                        - { name: geometry/scalewidth,
params: [770] }
                    post_processors:
                        jpegoptim: {}
```

Please refer to post-processors documentation in LiipImagineBundle for details.

## Drivers

LiipImagineBundle supports GD (default), Imagick and GMagick PHP extensions and only works on image blobs (no command line tool is needed). See the bundle's documentation to learn more on that topic.

## Upgrade

### Instantiate `LiipImagineBundle` in your kernel class

If you were using ImageMagick, install Imagick or Gmagick PHP extensions and activate the driver in `liip_imagine` (see LiipImagineBundle configuration documentation for more information):

```
# ezplatform.yml or config.yml
liip_imagine:
    # Driver can be either "imagick", "gmagick" or "gd",
depending on the PHP extension you're using.
    driver: imagick
```

> GD will be used by default if no driver is specified.

### Purging aliases

It is possible to use the Liip Imagine console tool to clear generated aliases.

```
$ php app/console liip:imagine:cache:remove
--filters=large
$ php app/console liip:imagine:cache:remove -v
```

The first example will clear the image files for the `large` alias. The second will clear all the generated aliases (be careful), and list the removed files (`-v`).

> The naming scheme change introduced by this feature wasn't enabled by default on 5.4.x. As part of migration you'll need to adapt to the new schema to get the benefit of this more efficient purge method. More technical information can be found on the pull request.

## Reference

### Available filters

In addition to filters exposed by LiipImagineBundle, the following are available:

| Filter name | Parameters | Description |
|---|---|---|
| geometry/scaledownonly | [width, height] | Generates a thumbnail that will not exceed width/height. |
| geometry/scalewidthdownonly | [width] | Generates a thumbnail that will not exceed width. |
| geometry/scaleheightdownonly | [height] | Generates a thumbnail that will not exceed height. |
| geometry/scalewidth | [width] | Alters image width. Proportion will be kept. |
| geometry/scaleheight | [height] | Alters image height. Proportion will be kept. |
| geometry/scale | [width, height] | Alters image size, not exceeding provided width and height. Proportion will be kept. |
| geometry/scaleexact | [width, height] | Alters image size to fit exactly provided width and height. Proportion will not be kept. |

| | | |
|---|---|---|
| geometry/scalepercent | [widthPercent, heightPercent] | Scales width and height with provided percent values. Proportion will not be kept. |
| geometry/crop | [width, height, startX, startY] | Crops the image. Result will have provided width/height, starting at provided startX/startY |
| border | [thickBorderX, thickBorderY, color=#000] | Adds a border around the image. Thickness is defined in px. Color is "#000" by default. |
| filter/noise | [radius=0] | Smooths the contours of an image (`imagick`/`gmagick` only). `radius` is in pixel. |
| filter/swirl | [degrees=60] | Swirls the pixels of the center of an image (`imagick`/`gmagick` only). `degrees` defaults to 60°. |
| resize | {size: [width, height]} | Simple resize filter (provided by LiipImagineBundle). |
| colorspace/gray | N/A | Converts an image to grayscale. |

> LiipImagineBundle supports additional settings, it is possible to combine filters from the list above to the ones provided in LiipImagineBundle or custom ones.

## Discarded filters

The following filters exist in the Imagine library but are not used in eZ Platform due to incompatibility:

- `flatten`. Obsolete, images are automatically flattened.
- `bordercolor`
- `border/width`
- `colorspace/transparent`
- `colorspace`

## Custom filters

Please refer to LiipImagineBundle documentation on custom filters. Imagine library documentation may also be useful.