Class upload

**What does it do?**

It manages file uploads for you. In short, it manages the uploaded file, and allows you to do whatever you want with the file, especially if it is an image, and as many times as you want.

It is the ideal class to quickly integrate file upload in your site. If the file is an image, you can convert, resize, crop it in many ways. You can also apply filters, add borders, text, watermarks, etc... That's all you need for a gallery script for instance. Supported formats are PNG, JPG, GIF and BMP.

You can also use the class to work on local files, which is especially useful to use the image manipulation features. The class also supports Flash uploaders.

The class works with PHP 4 and 5, and its error messages can be localized at will.

**How does it work?**

You instanciate the class with the $\_FILES['my\_field'] array where my\_field is the field name from your upload form. The class will check if the original file has been uploaded to its temporary location (alternatively, you can instanciate the class with a local filename).

You can then set a number of processing variables to act on the file. For instance, you can rename the file, and if it is an image, convert and resize it in many ways. You can also set what will the class do if the file already exists.

Then you call the function [process](http://www.verot.net/res/sources/class.upload.html#methodprocess) to actually perform the actions according to the processing parameters you set above. It will create new instances of the original file, so the original file remains the same between each process. The file will be manipulated, and copied to the given location. The processing variables will be reset once it is done.

You can repeat setting up a new set of processing variables, and calling [process](http://www.verot.net/res/sources/class.upload.html#methodprocess) again as many times as you want. When you have finished, you can call [clean](http://www.verot.net/res/sources/class.upload.html#methodclean) to delete the original uploaded file.

If you don't set any processing parameters and call [process](http://www.verot.net/res/sources/class.upload.html#methodprocess) just after instanciating the class. The uploaded file will be simply copied to the given location without any alteration or checks.

Don't forget to add *enctype="multipart/form-data"* in your form tag <form> if you want your form to upload the file.

**How to use it?**  
Create a simple HTML file, with a form such as:

<form enctype="multipart/form-data" method="post" action="upload.php">

<input type="file" size="32" name="image\_field" value="">

<input type="submit" name="Submit" value="upload">

</form>

Create a file called upload.php:

$handle = new upload($\_FILES['image\_field']);

if ($handle->uploaded) {

$handle->file\_new\_name\_body = 'image\_resized';

$handle->image\_resize = true;

$handle->image\_x = 100;

$handle->image\_ratio\_y = true;

$handle->process('/home/user/files/');

if ($handle->processed) {

echo 'image resized';

$handle->clean();

} else {

echo 'error : ' . $handle->error;

}

}

**How to process local files?**  
Use the class as following, the rest being the same as above:

$handle = new upload('/home/user/myfile.jpg');

**How to set the language?**  
Instantiate the class with a second argument being the language code:

$handle = new upload($\_FILES['image\_field'], 'fr\_FR');

$handle = new upload('/home/user/myfile.jpg', 'fr\_FR');

**How to output the resulting file or picture directly to the browser?**  
Simply call [process](http://www.verot.net/res/sources/class.upload.html#methodprocess)() without an argument (or with null as first argument):

$handle = new upload($\_FILES['image\_field']);

header('Content-type: ' . $handle->file\_src\_mime);

echo $handle->Process();

die();

Or if you want to force the download of the file:

$handle = new upload($\_FILES['image\_field']);

header('Content-type: ' . $handle->file\_src\_mime);

header("Content-Disposition: attachment; filename=".rawurlencode($handle->file\_src\_name).";");

echo $handle->Process();

die();

**Processing parameters** (reset after each process)

* [**file\_new\_name\_body**](http://www.verot.net/res/sources/class.upload.html#var$file_new_name_body) replaces the name body (default: '')

$handle->file\_new\_name\_body = 'new name';

* [**file\_name\_body\_add**](http://www.verot.net/res/sources/class.upload.html#var$file_name_body_add) appends to the name body (default: '')

$handle->file\_name\_body\_add = '\_uploaded';

* [**file\_name\_body\_pre**](http://www.verot.net/res/sources/class.upload.html#var$file_name_body_pre) prepends to the name body (default: '')

$handle->file\_name\_body\_pre = 'thumb\_';

* [**file\_new\_name\_ext**](http://www.verot.net/res/sources/class.upload.html#var$file_new_name_ext) replaces the file extension (default: '')

$handle->file\_new\_name\_ext = 'txt';

* [**file\_safe\_name**](http://www.verot.net/res/sources/class.upload.html#var$file_safe_name) formats the filename (spaces changed to \_) (default: true)

$handle->file\_safe\_name = true;

* [**file\_overwrite**](http://www.verot.net/res/sources/class.upload.html#var$file_overwrite) sets behaviour if file already exists (default: false)

$handle->file\_overwrite = true;

* [**file\_auto\_rename**](http://www.verot.net/res/sources/class.upload.html#var$file_auto_rename) automatically renames file if it already exists (default: true)

$handle->file\_auto\_rename = true;

* [**dir\_auto\_create**](http://www.verot.net/res/sources/class.upload.html#var$dir_auto_create) automatically creates destination directory if missing (default: true)

$handle->auto\_create\_dir = true;

* [**dir\_auto\_chmod**](http://www.verot.net/res/sources/class.upload.html#var$dir_auto_chmod) automatically attempts to chmod the destination directory if not writeable (default: true)

$handle->dir\_auto\_chmod = true;

* [**dir\_chmod**](http://www.verot.net/res/sources/class.upload.html#var$dir_chmod) chmod used when creating directory or if directory not writeable (default: 0777)

$handle->dir\_chmod = 0777;

* [**file\_max\_size**](http://www.verot.net/res/sources/class.upload.html#var$file_max_size) sets maximum upload size (default: upload\_max\_filesize from php.ini)

$handle->file\_max\_size = '1024'; // 1KB

* [**mime\_check**](http://www.verot.net/res/sources/class.upload.html#var$mime_check) sets if the class check the MIME against the [allowed](http://www.verot.net/res/sources/class.upload.html#var$allowed) list (default: true)

$handle->mime\_check = true;

* [**no\_script**](http://www.verot.net/res/sources/class.upload.html#var$no_script) sets if the class turns scripts into text files (default: true)

$handle->no\_script = false;

* [**allowed**](http://www.verot.net/res/sources/class.upload.html#var$allowed) array of allowed mime-types. wildcard accepted, as in image/\* (default: check Init)

$handle->allowed = array('application/pdf','application/msword', 'image/\*');

* [**forbidden**](http://www.verot.net/res/sources/class.upload.html#var$forbidden) array of forbidden mime-types. wildcard accepted, as in image/\* (default: check Init)

$handle->forbidden = array('application/\*');

* [**image\_convert**](http://www.verot.net/res/sources/class.upload.html#var$image_convert) if set, image will be converted (possible values : ''|'png'|'jpeg'|'gif'|'bmp'; default: '')

$handle->image\_convert = 'jpg';

* [**image\_background\_color**](http://www.verot.net/res/sources/class.upload.html#var$image_background_color) if set, will forcibly fill transparent areas with the color, in hexadecimal (default: null)

$handle->image\_background\_color = '#FF00FF';

* [**image\_default\_color**](http://www.verot.net/res/sources/class.upload.html#var$image_default_color) fallback color background color for non alpha-transparent output formats, such as JPEG or BMP, in hexadecimal (default: #FFFFFF)

$handle->image\_default\_color = '#FF00FF';

* [**jpeg\_quality**](http://www.verot.net/res/sources/class.upload.html#var$jpeg_quality) sets the compression quality for JPEG images (default: 85)

$handle->jpeg\_quality = 50;

* [**jpeg\_size**](http://www.verot.net/res/sources/class.upload.html#var$jpeg_size) if set to a size in bytes, will approximate [jpeg\_quality](http://www.verot.net/res/sources/class.upload.html#var$jpeg_quality) so the output image fits within the size (default: null)

$handle->jpeg\_size = 3072;

The following eight settings can be used to invalidate an upload if the file is an image (note that *open\_basedir* restrictions prevent the use of these settings)

* [**image\_max\_width**](http://www.verot.net/res/sources/class.upload.html#var$image_max_width) if set to a dimension in pixels, the upload will be invalid if the image width is greater (default: null)

$handle->image\_max\_width = 200;

* [**image\_max\_height**](http://www.verot.net/res/sources/class.upload.html#var$image_max_height) if set to a dimension in pixels, the upload will be invalid if the image height is greater (default: null)

$handle->image\_max\_height = 100;

* [**image\_max\_pixels**](http://www.verot.net/res/sources/class.upload.html#var$image_max_pixels) if set to a number of pixels, the upload will be invalid if the image number of pixels is greater (default: null)

$handle->image\_max\_pixels = 50000;

* [**image\_max\_ratio**](http://www.verot.net/res/sources/class.upload.html#var$image_max_ratio) if set to a aspect ratio (width/height), the upload will be invalid if the image apect ratio is greater (default: null)

$handle->image\_max\_ratio = 1.5;

* [**image\_min\_width**](http://www.verot.net/res/sources/class.upload.html#var$image_min_width) if set to a dimension in pixels, the upload will be invalid if the image width is lower (default: null)

$handle->image\_min\_width = 100;

* [**image\_min\_height**](http://www.verot.net/res/sources/class.upload.html#var$image_min_height) if set to a dimension in pixels, the upload will be invalid if the image height is lower (default: null)

$handle->image\_min\_height = 500;

* [**image\_min\_pixels**](http://www.verot.net/res/sources/class.upload.html#var$image_min_pixels) if set to a number of pixels, the upload will be invalid if the image number of pixels is lower (default: null)

$handle->image\_min\_pixels = 20000;

* [**image\_min\_ratio**](http://www.verot.net/res/sources/class.upload.html#var$image_min_ratio) if set to a aspect ratio (width/height), the upload will be invalid if the image apect ratio is lower (default: null)

$handle->image\_min\_ratio = 0.5;

* [**image\_resize**](http://www.verot.net/res/sources/class.upload.html#var$image_resize) determines is an image will be resized (default: false)

$handle->image\_resize = true;

The following variables are used only if [image\_resize](http://www.verot.net/res/sources/class.upload.html#var$image_resize) == true

* [**image\_x**](http://www.verot.net/res/sources/class.upload.html#var$image_x) destination image width (default: 150)

$handle->image\_x = 100;

* [**image\_y**](http://www.verot.net/res/sources/class.upload.html#var$image_y) destination image height (default: 150)

$handle->image\_y = 200;

Use either one of the following

* [**image\_ratio**](http://www.verot.net/res/sources/class.upload.html#var$image_ratio) if true, resize image conserving the original sizes ratio, using [image\_x](http://www.verot.net/res/sources/class.upload.html#var$image_x) AND [image\_y](http://www.verot.net/res/sources/class.upload.html#var$image_y) as max sizes if true (default: false)

$handle->image\_ratio = true;

* [**image\_ratio\_crop**](http://www.verot.net/res/sources/class.upload.html#var$image_ratio_crop) if true, resize image conserving the original sizes ratio, using [image\_x](http://www.verot.net/res/sources/class.upload.html#var$image_x) AND [image\_y](http://www.verot.net/res/sources/class.upload.html#var$image_y) as max sizes, and cropping excedent to fill the space. setting can also be a string, with one or more from 'TBLR', indicating which side of the image will be kept while cropping (default: false)

$handle->image\_ratio\_crop = true;

* [**image\_ratio\_fill**](http://www.verot.net/res/sources/class.upload.html#var$image_ratio_fill) if true, resize image conserving the original sizes ratio, using [image\_x](http://www.verot.net/res/sources/class.upload.html#var$image_x) AND [image\_y](http://www.verot.net/res/sources/class.upload.html#var$image_y) as max sizes, fitting the image in the space and coloring the remaining space. setting can also be a string, with one or more from 'TBLR', indicating which side of the space the image will be in (default: false)

$handle->image\_ratio\_fill = true;

* [**image\_ratio\_no\_zoom\_in**](http://www.verot.net/res/sources/class.upload.html#var$image_ratio_no_zoom_in) same as [image\_ratio](http://www.verot.net/res/sources/class.upload.html#var$image_ratio), but won't resize if the source image is smaller than [image\_x](http://www.verot.net/res/sources/class.upload.html#var$image_x) x [image\_y](http://www.verot.net/res/sources/class.upload.html#var$image_y) (default: false)

$handle->image\_ratio\_no\_zoom\_in = true;

* [**image\_ratio\_no\_zoom\_out**](http://www.verot.net/res/sources/class.upload.html#var$image_ratio_no_zoom_out) same as [image\_ratio](http://www.verot.net/res/sources/class.upload.html#var$image_ratio), but won't resize if the source image is bigger than [image\_x](http://www.verot.net/res/sources/class.upload.html#var$image_x) x [image\_y](http://www.verot.net/res/sources/class.upload.html#var$image_y) (default: false)

$handle->image\_ratio\_no\_zoom\_out = true;

* [**image\_ratio\_x**](http://www.verot.net/res/sources/class.upload.html#var$image_ratio_x) if true, resize image, calculating [image\_x](http://www.verot.net/res/sources/class.upload.html#var$image_x) from [image\_y](http://www.verot.net/res/sources/class.upload.html#var$image_y) and conserving the original sizes ratio (default: false)

$handle->image\_ratio\_x = true;

* [**image\_ratio\_y**](http://www.verot.net/res/sources/class.upload.html#var$image_ratio_y) if true, resize image, calculating [image\_y](http://www.verot.net/res/sources/class.upload.html#var$image_y) from [image\_x](http://www.verot.net/res/sources/class.upload.html#var$image_x) and conserving the original sizes ratio (default: false)

$handle->image\_ratio\_y = true;

* [**image\_ratio\_pixels**](http://www.verot.net/res/sources/class.upload.html#var$image_ratio_pixels) if set to a long integer, resize image, calculating [image\_y](http://www.verot.net/res/sources/class.upload.html#var$image_y) and [image\_x](http://www.verot.net/res/sources/class.upload.html#var$image_x) to match a the number of pixels (default: false)

$handle->image\_ratio\_pixels = 25000;

The following image manipulations require GD2+

* [**image\_brightness**](http://www.verot.net/res/sources/class.upload.html#var$image_brightness) if set, corrects the brightness. value between -127 and 127 (default: null)

$handle->image\_brightness = 40;

* [**image\_contrast**](http://www.verot.net/res/sources/class.upload.html#var$image_contrast) if set, corrects the contrast. value between -127 and 127 (default: null)

$handle->image\_contrast = 50;

* [**image\_tint\_color**](http://www.verot.net/res/sources/class.upload.html#var$image_tint_color) if set, will tint the image with a color, value as hexadecimal #FFFFFF (default: null)

$handle->image\_tint\_color = '#FF0000';

* [**image\_overlay\_color**](http://www.verot.net/res/sources/class.upload.html#var$image_overlay_color) if set, will add a colored overlay, value as hexadecimal #FFFFFF (default: null)

$handle->image\_overlay\_color = '#FF0000';

* [**image\_overlay\_percent**](http://www.verot.net/res/sources/class.upload.html#var$image_overlay_percent) used when [image\_overlay\_color](http://www.verot.net/res/sources/class.upload.html#var$image_overlay_color) is set, determines the opacity (default: 50)

$handle->image\_overlay\_percent = 20;

* [**image\_negative**](http://www.verot.net/res/sources/class.upload.html#var$image_negative) inverts the colors in the image (default: false)

$handle->image\_negative = true;

* [**image\_greyscale**](http://www.verot.net/res/sources/class.upload.html#var$image_greyscale) transforms an image into greyscale (default: false)

$handle->image\_greyscale = true;

* [**image\_threshold**](http://www.verot.net/res/sources/class.upload.html#var$image_threshold) applies a threshold filter. value between -127 and 127 (default: null)

$handle->image\_threshold = 20;

* [**image\_unsharp**](http://www.verot.net/res/sources/class.upload.html#var$image_unsharp) applies an unsharp mask, with alpha transparency support (default: false)

$handle->image\_unsharp = true;

* [**image\_unsharp\_amount**](http://www.verot.net/res/sources/class.upload.html#var$image_unsharp_amount) unsharp mask amount, typically 50 - 200 (default: 80)

$handle->image\_unsharp\_amount = 120;

* [**image\_unsharp\_radius**](http://www.verot.net/res/sources/class.upload.html#var$image_unsharp_radius) unsharp mask radius, typically 0.5 - 1 (default: 0.5)

$handle->image\_unsharp\_radius = 0.8;

* [**image\_unsharp\_threshold**](http://www.verot.net/res/sources/class.upload.html#var$image_unsharp_threshold) unsharp mask threshold, typically 0 - 5 (default: 1)

$handle->image\_unsharp\_threshold = 0;

* [**image\_text**](http://www.verot.net/res/sources/class.upload.html#var$image_text) creates a text label on the image, value is a string, with eventual replacement tokens (default: null)

$handle->image\_text = 'test';

* [**image\_text\_direction**](http://www.verot.net/res/sources/class.upload.html#var$image_text_direction) text label direction, either 'h' horizontal or 'v' vertical (default: 'h')

$handle->image\_text\_direction = 'v';

* [**image\_text\_color**](http://www.verot.net/res/sources/class.upload.html#var$image_text_color) text color for the text label, in hexadecimal (default: #FFFFFF)

$handle->image\_text\_color = '#FF0000';

* [**image\_text\_percent**](http://www.verot.net/res/sources/class.upload.html#var$image_text_percent) text opacity on the text label, integer between 0 and 100 (default: 100)

$handle->image\_text\_percent = 50;

* [**image\_text\_background**](http://www.verot.net/res/sources/class.upload.html#var$image_text_background) text label background color, in hexadecimal (default: null)

$handle->image\_text\_background = '#FFFFFF';

* [**image\_text\_background\_percent**](http://www.verot.net/res/sources/class.upload.html#var$image_text_background_percent) text label background opacity, integer between 0 and 100 (default: 100)

$handle->image\_text\_background\_percent = 50;

* [**image\_text\_font**](http://www.verot.net/res/sources/class.upload.html#var$image_text_font) built-in font for the text label, from 1 to 5. 1 is the smallest (default: 5)

$handle->image\_text\_font = 4;

* [**image\_text\_x**](http://www.verot.net/res/sources/class.upload.html#var$image_text_x) absolute text label position, in pixels from the left border. can be negative (default: null)

$handle->image\_text\_x = 5;

* [**image\_text\_y**](http://www.verot.net/res/sources/class.upload.html#var$image_text_y) absolute text label position, in pixels from the top border. can be negative (default: null)

$handle->image\_text\_y = 5;

* [**image\_text\_position**](http://www.verot.net/res/sources/class.upload.html#var$image_text_position) text label position withing the image, a combination of one or two from 'TBLR': top, bottom, left, right (default: null)

$handle->image\_text\_position = 'LR';

* [**image\_text\_padding**](http://www.verot.net/res/sources/class.upload.html#var$image_text_padding) text label padding, in pixels. can be overridden by [image\_text\_padding\_x](http://www.verot.net/res/sources/class.upload.html#var$image_text_padding_x) and [image\_text\_padding\_y](http://www.verot.net/res/sources/class.upload.html#var$image_text_padding_y) (default: 0)

$handle->image\_text\_padding = 5;

* [**image\_text\_padding\_x**](http://www.verot.net/res/sources/class.upload.html#var$image_text_padding_x) text label horizontal padding (default: null)

$handle->image\_text\_padding\_x = 2;

* [**image\_text\_padding\_y**](http://www.verot.net/res/sources/class.upload.html#var$image_text_padding_y) text label vertical padding (default: null)

$handle->image\_text\_padding\_y = 10;

* [**image\_text\_alignment**](http://www.verot.net/res/sources/class.upload.html#var$image_text_alignment) text alignment when text has multiple lines, either 'L', 'C' or 'R' (default: 'C')

$handle->image\_text\_alignment = 'R';

* [**image\_text\_line\_spacing**](http://www.verot.net/res/sources/class.upload.html#var$image_text_line_spacing) space between lines in pixels, when text has multiple lines (default: 0)

$handle->image\_text\_line\_spacing = 3;

* [**image\_flip**](http://www.verot.net/res/sources/class.upload.html#var$image_flip) flips image, wither 'h' horizontal or 'v' vertical (default: null)

$handle->image\_flip = 'h';

* [**image\_rotate**](http://www.verot.net/res/sources/class.upload.html#var$image_rotate) rotates image. possible values are 90, 180 and 270 (default: null)

$handle->image\_rotate = 90;

* [**image\_crop**](http://www.verot.net/res/sources/class.upload.html#var$image_crop) crops image. accepts 4, 2 or 1 values as 'T R B L' or 'TB LR' or 'TBLR'. dimension can be 20, or 20px or 20% (default: null)

$handle->image\_crop = array(50,40,30,20); OR '-20 20%'...

* [**image\_precrop**](http://www.verot.net/res/sources/class.upload.html#var$image_precrop) crops image, before an eventual resizing. accepts 4, 2 or 1 values as 'T R B L' or 'TB LR' or 'TBLR'. dimension can be 20, or 20px or 20% (default: null)

$handle->image\_precrop = array(50,40,30,20); OR '-20 20%'...

* [**image\_bevel**](http://www.verot.net/res/sources/class.upload.html#var$image_bevel) adds a bevel border to the image. value is thickness in pixels (default: null)

$handle->image\_bevel = 20;

* [**image\_bevel\_color1**](http://www.verot.net/res/sources/class.upload.html#var$image_bevel_color1) top and left bevel color, in hexadecimal (default: #FFFFFF)

$handle->image\_bevel\_color1 = '#FFFFFF';

* [**image\_bevel\_color2**](http://www.verot.net/res/sources/class.upload.html#var$image_bevel_color2) bottom and right bevel color, in hexadecimal (default: #000000)

$handle->image\_bevel\_color2 = '#000000';

* [**image\_border**](http://www.verot.net/res/sources/class.upload.html#var$image_border) adds a unicolor border to the image. accepts 4, 2 or 1 values as 'T R B L' or 'TB LR' or 'TBLR'. dimension can be 20, or 20px or 20% (default: null)

$handle->image\_border = '3px'; OR '-20 20%' OR array(3,2)...

* [**image\_border\_color**](http://www.verot.net/res/sources/class.upload.html#var$image_border_color) border color, in hexadecimal (default: #FFFFFF)

$handle->image\_border\_color = '#FFFFFF';

* [**image\_frame**](http://www.verot.net/res/sources/class.upload.html#var$image_frame) type of frame: 1=flat 2=crossed (default: null)

$handle->image\_frame = 2;

* [**image\_frame\_colors**](http://www.verot.net/res/sources/class.upload.html#var$image_frame_colors) list of hex colors, in an array or a space separated string (default: '#FFFFFF #999999 #666666 #000000')

$handle->image\_frame\_colors = array('#999999', '#FF0000', '#666666', '#333333', '#000000');

* [**image\_watermark**](http://www.verot.net/res/sources/class.upload.html#var$image_watermark) adds a watermark on the image, value is a local filename. accepted files are GIF, JPG, BMP, PNG and PNG alpha (default: null)

$handle->image\_watermark = 'watermark.png';

* [**image\_watermark\_x**](http://www.verot.net/res/sources/class.upload.html#var$image_watermark_x) absolute watermark position, in pixels from the left border. can be negative (default: null)

$handle->image\_watermark\_x = 5;

* [**image\_watermark\_y**](http://www.verot.net/res/sources/class.upload.html#var$image_watermark_y) absolute watermark position, in pixels from the top border. can be negative (default: null)

$handle->image\_watermark\_y = 5;

* [**image\_watermark\_position**](http://www.verot.net/res/sources/class.upload.html#var$image_watermark_position) watermark position withing the image, a combination of one or two from 'TBLR': top, bottom, left, right (default: null)

$handle->image\_watermark\_position = 'LR';

* [**image\_watermark\_no\_zoom\_in**](http://www.verot.net/res/sources/class.upload.html#var$image_watermark_no_zoom_in) prevents the watermark to be resized up if it is smaller than the image (default: true)

$handle->image\_watermark\_no\_zoom\_in = false;

* [**image\_watermark\_no\_zoom\_out**](http://www.verot.net/res/sources/class.upload.html#var$image_watermark_no_zoom_out) prevents the watermark to be resized down if it is bigger than the image (default: false)

$handle->image\_watermark\_no\_zoom\_out = true;

* [**image\_reflection\_height**](http://www.verot.net/res/sources/class.upload.html#var$image_reflection_height) if set, a reflection will be added. Format is either in pixels or percentage, such as 40, '40', '40px' or '40%' (default: null)

$handle->image\_reflection\_height = '25%';

* [**image\_reflection\_space**](http://www.verot.net/res/sources/class.upload.html#var$image_reflection_space) space in pixels between the source image and the reflection, can be negative (default: null)

$handle->image\_reflection\_space = 3;

* [**image\_reflection\_color**](http://www.verot.net/res/sources/class.upload.html#var$image_reflection_color) reflection background color, in hexadecimal. Now deprecated in favor of [image\_default\_color](http://www.verot.net/res/sources/class.upload.html#var$image_default_color) (default: #FFFFFF)

$handle->image\_default\_color = '#000000';

* [**image\_reflection\_opacity**](http://www.verot.net/res/sources/class.upload.html#var$image_reflection_opacity) opacity level at which the reflection starts, integer between 0 and 100 (default: 60)

$handle->image\_reflection\_opacity = 60;

**Values that can be read before calling** [**process**](http://www.verot.net/res/sources/class.upload.html#methodprocess)**()**

* [**file\_src\_name**](http://www.verot.net/res/sources/class.upload.html#var$file_src_name) Source file name
* [**file\_src\_name\_body**](http://www.verot.net/res/sources/class.upload.html#var$file_src_name_body) Source file name body
* [**file\_src\_name\_ext**](http://www.verot.net/res/sources/class.upload.html#var$file_src_name_ext) Source file extension
* [**file\_src\_pathname**](http://www.verot.net/res/sources/class.upload.html#var$file_src_pathname) Source file complete path and name
* [**file\_src\_mime**](http://www.verot.net/res/sources/class.upload.html#var$file_src_mime) Source file mime type
* [**file\_src\_size**](http://www.verot.net/res/sources/class.upload.html#var$file_src_size) Source file size in bytes
* [**file\_src\_error**](http://www.verot.net/res/sources/class.upload.html#var$file_src_error) Upload error code
* [**file\_is\_image**](http://www.verot.net/res/sources/class.upload.html#var$file_is_image) Boolean flag, true if the file is a supported image type

If the file is a supported image type (and *open\_basedir* restrictions allow it)

* [**image\_src\_x**](http://www.verot.net/res/sources/class.upload.html#var$image_src_x) Source file width in pixels
* [**image\_src\_y**](http://www.verot.net/res/sources/class.upload.html#var$image_src_y) Source file height in pixels
* [**image\_src\_pixels**](http://www.verot.net/res/sources/class.upload.html#var$image_src_pixels) Source file number of pixels
* [**image\_src\_type**](http://www.verot.net/res/sources/class.upload.html#var$image_src_type) Source file type (png, jpg, gif or bmp)
* [**image\_src\_bits**](http://www.verot.net/res/sources/class.upload.html#var$image_src_bits) Source file color depth

**Values that can be read after calling** [**process**](http://www.verot.net/res/sources/class.upload.html#methodprocess)**()**

* [**file\_dst\_path**](http://www.verot.net/res/sources/class.upload.html#var$file_dst_path) Destination file path
* [**file\_dst\_name\_body**](http://www.verot.net/res/sources/class.upload.html#var$file_dst_name_body) Destination file name body
* [**file\_dst\_name\_ext**](http://www.verot.net/res/sources/class.upload.html#var$file_dst_name_ext) Destination file extension
* [**file\_dst\_name**](http://www.verot.net/res/sources/class.upload.html#var$file_dst_name) Destination file name
* [**file\_dst\_pathname**](http://www.verot.net/res/sources/class.upload.html#var$file_dst_pathname) Destination file complete path and name

If the file is a supported image type

* [**image\_dst\_x**](http://www.verot.net/res/sources/class.upload.html#var$image_dst_x) Destination file width
* [**image\_dst\_y**](http://www.verot.net/res/sources/class.upload.html#var$image_dst_y) Destination file height
* [**image\_convert**](http://www.verot.net/res/sources/class.upload.html#var$image_convert) Destination file format