

# Transforming Image Files

Last Modified on 03/18/2017 7:50 pm EDT

## Overview

Salsify supports an extensive library of digital asset transformations as part of configuring your channel or catalog's digital asset export. You can resize and crop images, apply watermarks, and much more. You can also apply transformations to video files. [Click here](#) for more information on video transformations.

**Image Format**

**Transform Images**

Optionally provide any instructions for transforming each image, including resizing, cropping, or padding. By default, each image will simply be copied as-is.

[Download a Sample](#)

## How Image Transformations Work

The image URL contains the instructions for the transformation. They are combined in a comma-delimited string as part of the URL. In this example, the transformation section is `/w_100,h_150,c_scale`.

Each transformation has two parts, the parameter and the value, separated by an underscore, and each pair is separated by a comma. So in our example, we are applying three transformations. The width is being changed to 100, the height is being changed to 150, and a basic default scale is applied. You can see in the example, the original image was more wide than square, so it's being distorted horizontally with the transformation.

Transformations are combined into a comma-delimited string of transformation codes. For example, scaling the sample image (originally 864x576) to **100x150**:

[http://a1.images.salsify.com/image/upload/w\\_100,h\\_150,c\\_scale/sample.jpg](http://a1.images.salsify.com/image/upload/w_100,h_150,c_scale/sample.jpg)



The `w_100,h_150,c_scale` instructs the system to set the width to 100, height to 150, and use `scale` as the crop mode. You can modify the URL (as above) or click 'Download a Sample' to test our your transformation.

For example, if you remove the transformation string from the URL and paste the resulting URL into your browser, you'll see the original size.

<http://a1.images.salsify.com/image/upload/sample.jpg>

You can apply the transformations in the table below to the example image to see how it changes.

See the Reference section below for all of the available options, and Common Transformation Examples for other examples of uses.

## Image Transformations Reference

Click on the examples to open the image in a new window and view the image URL transformations that have been applied.

Param	Value	Example	Description
<code>c</code>	<code>mode</code>		<p>A crop mode that determines how to transform the image for fitting into the desired width &amp; height dimensions.</p> <p>Extract a region of the given width and height out of the</p>

original image.  
The original proportions are retained and so is the size of the graphics. You can specify the gravity parameter to select which part of the image to extract, or use fixed coordinates cropping.

---

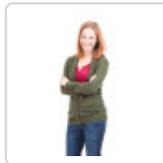
*scale(default)*



Change the size of the image to match the given width & height. All original image parts will be visible but might be stretched or shrunken.

---

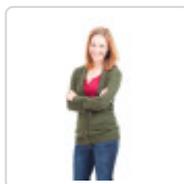
*fill*



Create an image with the exact given width and height while retaining original proportions. Uses only a portion of the original image that fills the given dimensions.

---

*lfill*



Same as the 'fill' mode but doesn't expand the image if your requested dimensions are

larger than the original image's.

*fit*



Change image size to fit in the given width & height while retaining original proportions. All original image parts are visible. Both width and height dimensions of the transformed image must not exceed the specified width & height.

*mfit*



Scale the image up to fit the given width & height while retaining original proportions.

*limit*



Used for creating an image that does not exceed the given width or height.

*pad*



Resize the image to fill the given width & height while retaining original proportions. Padding will be added if the original image proportions do not match the required ones.

*lpad*



Same as the 'pad' mode but doesn't scale the image up if your requested dimensions are larger than the original image's.

*mpad*



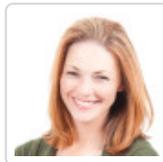
Same as the 'pad' mode but doesn't scale the original image.

*crop*



Used to extract a given width & height out of the original image. The original proportions are retained and so is the size of the graphics.

*thumb*



Generate a thumbnail using face detection in combination with the 'face' or 'faces' gravity.

**w**

*pixels or percents*

The required width of a transformed image or an overlay. Can be specified separately or together with the height value.

80



Resize width to 80 pixels while maintaining aspect ratio.

0.2



Resize image to 20% of its original size.

**h**

*pixels or percents*

The required height of a transformed image or an overlay. Can be specified separately or together with the width value.

40



Resize height to 40 pixels while maintaining aspect ratio.

0.3



Resize image to 30% of its original size.

**ar**

*a:b ratio (eg 16:9) or*

Used with a crop mode (scale, fill,

decimal representing the ratio of the width divided by the height (e.g., 1.33 or 2.5)

lfill, pad, lpad, mpad or crop) to determine how the image is adjusted to the new dimensions. Can also be used with either width or height to create a proportional transform. The other dimension is then automatically updated to maintain the given aspect ratio.

---

16:9



Sized to width = 80, 16:9 ratio height

---

1.2



Sized to width = 80, 1.2 ratio height

---

**z**

percent

Control how much of the original image surrounding the face to keep when using either the 'crop' or 'thumb' cropping modes with face detection (Default: 1.0).

	1.2		Cropped with face detection and zoom set to 120%.
--	-----	---	---

	.75		T thumbnail with face detection and zoom set to 75%.
--	-----	---	--

sg	<i>direction</i>		Decides which part of the image to keep while 'crop', 'pad' and 'fill' crop modes are used. For overlays, this decides where to place the overlay.
----	------------------	--	--

	<i>north_west</i>		North west corner (top left).
--	-------------------	---	-------------------------------

	<i>north</i>		North center part (top center).
--	--------------	---	---------------------------------

	<i>north_east</i>		North east corner (top right).
--	-------------------	---	--------------------------------

<i>west</i>		Middle west part (left).
<i>center(default)</i>		The center of the image.
<i>east</i>		Middle east part (right).
<i>south_west</i>		South west corner (bottom left).
<i>south</i>		South center part (bottom center).
<i>south_east</i>		South east corner (bottom right).
<i>xy_center</i>		Set the crop's center of gravity to the given x & y coordinates"
<i>face</i>		Automatically



detects the largest face in an image and aim to make it the center of the cropped image.

*face (thumb)*



Automatically detects the largest face in an image and use it to generate a face thumbnail.

*faces*



Automatically detect multiple faces in an image and aim to make them the center of the cropped image.

*face:center*



Same as the 'face' gravity, but with fallback to 'center' gravity instead of 'north' if no face is detected.

*faces:center*



Same as the 'faces' gravity, but with fallback to 'center' gravity instead of 'north' if no face is detected.

**x**

*pixels*

Horizontal position for custom-coordinates based cropping, overlay placement and

certain region related effects.

130



Crop image to an 80x80 square starting 130 pixels from the left.

**y**

*pixels*

Vertical position for custom-coordinates based cropping and overlay placement.

340



Crop image to an 80x80 square starting 340 pixels from the top.

**q**

*percents*

Control the JPG compression quality. 1 is the lowest quality and 100 is the highest. The default is the original image's quality or 90% if not available. Reducing quality generates JPG images much smaller in file size.

100

Generate a thumbnail using highest image



quality (9.15KB)

20



Generate a thumbnail using a low 20% quality (1.48KB)

**r**

*pixels or max*

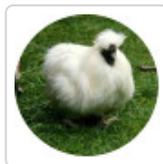
Round the corners of an image or make it completely circular or oval (ellipse).

20



Generate a small image with rounded corners of 20 pixels radius.

*max*



Generate an image with a circular crop using the 'max' radius value.

**a**

*degrees or mode*

Rotate or flip an image by the given degrees or automatically according to its orientation or available meta-data. Multiple modes can be

applied by concatenating their names with a dot.

90



Rotate image by 90 degrees clockwise.

10



Rotate image by 10 degrees clockwise.

-20



Rotate image by 20 degrees counter-clockwise.

*auto\_right*



Rotate image 90 degrees clockwise only if the requested aspect ratio does not match the image's aspect ratio.

*auto\_left*



Rotate image 90 degrees counter-clockwise only if the requested aspect ratio does not match the image's aspect ratio.

*auto*



Same as `auto_right`. Rotate image 90 degrees clockwise only if the requested aspect ratio does not match the image's aspect ratio.

*exif*



Automatically rotate the image according to the EXIF data stored by the camera when the image was taken.

*vflip*



Vertical mirror flip of the image.

*hflip*

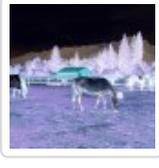


Horizontal mirror flip of the image.

**e**

*name and value*

Apply a filter or an effect on an image. The value includes the name of the effect and an additional parameter that controls the behavior of the

<i>grayscale</i>		specific effect. Convert image to gray-scale (multiple shades of gray).
<i>blackwhite</i>		Covert image to black and white.
<i>oil_paint</i>		Apply an oil painting effect.
<i>negate</i>		Negate image colors (negative).
<i>vignette</i>		Apply a vignette effect.
<i>sepia</i>		Change the color scheme of the image to sepia. (default level: 80).
<i>sepia:50</i>		Apply the sepia effect with strength of '50'.

*brightness:60*



Increase image  
brightness by  
60% (default:  
80)

*brightness:-40*



Decrease image  
brightness by  
40%.

*auto\_brightness*



Automatically  
adjust  
brightness.

*fill\_light*



Adjust the fill  
light of an image.  
(level range: -  
100-100, default  
value: 0)

*saturation:70*



Increase the  
image's color  
saturation by  
70% (default:  
80).

*saturation:-50*



Decrease the  
image's color  
saturation by  
50%.

*hue:40*

Alter the image's  
hue by 40  
(default: 80).



*shadow:50*



Add a green shadow with a strength of 50 and with an offset specified by x:-3 & y:3.

*pixelate*



Apply a pixelate effect on the image.

*pixelate:3*



Apply a pixelate effect using 3 pixels wide pixelation squares.

*pixelate\_region*



Pixelate only a certain region of the image based on the given x, y, width and height.

*pixelate\_region:20*



Pixelate only a certain region of the image using 20 pixels wide pixelation squares.

*pixelate\_faces*

Automatically pixelate all detected faces in the image.

<i>pixelate_faces:3</i>		Automatically pixelate all detected faces in the image using 3 pixels wide pixelation squares.
<i>redeye</i>		Automatically remove red eyes in an image.
<i>gradient_fade</i>		Apply a gradient fade effect on the image.
<i>blur</i>		Apply a blurring filter on the image. (level range: 1-2000, default level: 100)
<i>blur:300</i>		Apply a strong blurring filter of level 300 on the image.
<i>blur_region</i>		Apply a blurring filter on a certain region of an image, specified by x, y, width and height. (level range: 1-2000, default level: 100)
<i>blur_faces</i>		Automatically blur all detected



faces in the image. (level range: 1-2000, default level: 100)

---

*distort*



Distorts the image to a new shape with coordinates 5:34:70:10:70:7 5:5:55

---

*sheer:20:0*



Skews the image according to two specified values in degrees separated by a colon (:), representing how much to skew the image on the x-axis and y-axis respectively. Negative values skew the image in the opposite direction. Skews the image on the x-axis by 20 degrees.

---

*sharpen*



Sharpen the image. (level range: 1-2000, default level: 100)

---

*sharpen:400*



Apply a strong sharpening filter of level 400.

<i>unsharp_mask</i>		Sharpen the image using the unsharp mask filter. (level range: 1-2000, default level: 100)
<i>unsharp_mask:400</i>		Apply a strong unsharp mask filter of level 400.
<i>contrast:90</i>		Increase image contrast by 90%.
<i>contrast:-70</i>		Decrease image contrast by 70%.
<i>auto_contrast</i>		Automatically adjust contrast.
<i>vibrance</i>		Apply a vibrance filter on the image. (level range: -100-100, default level: 20)
<i>vibrance:70</i>		Apply a strong vibrance filter of level 70 on the image.

*red:50*



Strengthen the image's red channel by 50%.

*green:50*



Strengthen the image's green channel by 50%.

*blue:90*



Strengthen the image's blue channel by 90%.

*auto\_color*



Automatically adjust color balance.

*improve*



Automatically adjust image colors, contrast and lightness.

*screen*



Add an overlay image blended using the 'screen' blend mode. In this mode, each pixel of the image is made brighter

according to the pixel value of the overlaid image.

---

*multiply*



Add an overlay image blended using the 'screen' blend mode. In this mode, each pixel of the image is made darker according to the pixel value of the overlaid image.

---

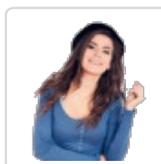
*overlay*



Add an overlay image blended using the 'overlay' blend mode. In this mode, each pixel of the image is made darker or brighter according to the pixel value of the overlaid image.

---

*make\_transparent*



Make the background of the image transparent (or solid white for JPGs). The background is determined as all pixels that resemble the pixels in the image's edges. (level range: 0-100, default level: 10)

---

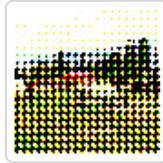
*trim*

Trim solid pixels from image



edges. (level range: 0-100, default level: 10)

*ordered\_dither*



Apply an ordered dither filter on the image. Possible levels described below (Default 0):

- |           |  |
|-----------|--|
| <b>0</b>  | <b>Threshold<br/>1x1 (non-dither)</b>      |
| <b>1</b>  | <b>Checkerboard<br/>2x1 (dither)</b>       |
| <b>2</b>  | <b>Ordered 2x2<br/>(dispersed)</b>         |
| <b>3</b>  | <b>Ordered 3x3<br/>(dispersed)</b>         |
| <b>4</b>  | <b>Ordered 4x4<br/>(dispersed)</b>         |
| <b>5</b>  | <b>Ordered 8x8<br/>(dispersed)</b>         |
| <b>6</b>  | <b>Halftone 4x4<br/>(angled)</b>           |
| <b>7</b>  | <b>Halftone 6x6<br/>(angled)</b>           |
| <b>8</b>  | <b>Halftone 8x8<br/>(angled)</b>           |
| <b>9</b>  | <b>Halftone 4x4<br/>(orthogonal)</b>       |
| <b>10</b> | <b>Halftone 6x6<br/>(orthogonal)</b>       |
| <b>11</b> | <b>Halftone 8x8<br/>(orthogonal)</b>       |
| <b>12</b> | <b>Halftone<br/>16x16<br/>(orthogonal)</b> |
| <b>13</b> | <b>Circles 5x5<br/>(black)</b>             |
| <b>14</b> | <b>Circles 5x5<br/>(white)</b>             |
| <b>15</b> | <b>Circles 6x6<br/>(black)</b>             |
| <b>16</b> | <b>Circles 6x6<br/>(white)</b>             |
| <b>17</b> | <b>Circles 7x7<br/>(black)</b>             |

**o***percents*

Adjust the opacity of the image and make it semi-transparent. 100 means opaque, while 0 is completely transparent.

40



Reduce image opacity to 40%.

**bo***style*

Add a solid border around the image. The value has a CSS-like format: width\_style\_color.

*5px\_solid\_black*

Add a 5 pixels wide black border to the image.

*4px\_solid\_rgb:00390b*

Add a 4 pixels wide border of the color #00390b.

*5px\_solid\_rgb:00390*

Add a 5 pixels

*b60*



wide border of a semi transparent RGB color. The last 2 digits are the hex value of the alpha channel.

---

**b**

*color*

Sets the background color to use instead of transparent background areas when converting to JPG format or using the pad crop mode. The background color can be set as an RGB hex triplet (e.g. 'b\_rgb:3e2222'), a 3 character RGB hex (e.g. 'b\_rgb:777') or a named color (e.g. 'b\_green').

---

**if**

*condition*

Apply a transformation only if a specified condition is met. For use in combination with other transformations.

---

*if\_else*

Add to specify transformation where if\_ condition is not

met.

---

<b>cs</b>	<i>color space</i>		Sets the color space for an image to change it to or from web-friendly sRGB or print-standard CMYK.
	<i>rgb</i>		Changes to sRGB color space.
	<i>cmk</i>		Changes to CMYK color space.

---

<b>I</b>	<i>identifier</i>		Add an overlay image over the base image. You can control the dimension and position of the overlay using the width, height, x, y and gravity parameters. Please contact us for more details on adding an overlay image.
----------	-------------------	--	--

---

*tbadge*



Add the overlay with the ID 'tbadge' 10, 20 pixels from the south east corner of the base image while resizing the

overlay to have a width of 30 pixels.

# Common Transformation Examples

`dn_300,cs_srgb` Convert to 300dpi and ensure web-friendly color space

`c_fit,w_3500,h_3500,dn_300,cs_srgb` Fit to 3500x3500 while maintaining aspect ratio, convert to 300dpi and web-friendly color space

`c_pad,w_2000,h_2000,dn_300,cs_srgb` Pad to fit 2000x2000 while maintaining aspect ratio and converting to web-friendly color space

`c_fill,ar_1:1` Crop image to fit a 1x1 box, will take the center of the image unless you add gravity

`if_w_lt_1000,c_mfit,dn_300,w_1000,h_1000/if_else,c_fit,w_2000,h_2000` If image is less than 1000px wide, upscale to 1000x1000, otherwise fit to 2000x2000

`e_clip` Trim pixels according to a clipping path included in the original image metadata (e.g., manually created using software such as Adobe PhotoShop).

`e_trim` Trim pixels from the border of your image. Helpful to add a specific border size.

`q_50` Significantly reduces the quality of your imagery for smaller files, often without compromising the visual on a webpage. Default is 90, resulting in a larger file. Test and reduce for web use to reduce image load times.