SPEEDSIZE™ TECHNICAL GUIDE

This document expands on the use of AI Compression, Adaptive Delivery and URL Parameters to optimize the effectiveness and value of SpeedSize™

SPEEDSIZE™

AI-OPTIMIZE YOUR WEBSITE IMAGES & VIDEOS

Load your website images & videos up to 70% faster with 100% original quality

SpeedSize.com is the most advanced alternative to conventional compression and delivery, no-code AI-driven platform delivers the fastest visual experience in the market.

SPEEDSIZE™ VALUE

- Markets Highest Visual Quality
- Crushes Leading Images & Videos
- Enhances Customers Buying Journey
- Plug & Play Integration
- Quickly Achieving Short Videos
- Optimizing Media Workflows
- Improves Core Web Vitals & SEO
- Global Edge CDN With 600+ Locations
- Device and Bandwidth Adaptive Delivery
- Amplifies Brand Cue & Background Personal
- Transformations For Image Editing
- AI-powered Upscale & Quality Enhancement
- CO2e Reduction By Delivering Less Data
- Prevent Human Errors At Scale

INTEGRATION

SpeedSize™ is proud to have the easiest and fastest integration in the industry. With plugins for Magento, Shopify and WordPress/WooCommerce, integration is quick and seamless. For a custom-built website and integration, simply add a "lazy-load" script or use "website.com" custom subdomain prefix to your media URLs – it's that easy.

- **Plugins**
- **Link**
- **Script**
- **API**
- **Interface**

LINK STRUCTURE

Rewrite your existing media links to SpeedSize™ to AI-optimize your images and videos seamlessly

```
https://acme.speedsize.com/cross/your-website.com/img/cover.jpg?powerbox=w_200,h_100
```

CUSTOM SUBDOMAIN / CLIENT_ID	LINK	OPTIONAL PARAMS

CUSTOM SUBDOMAIN / CLIENT_ID
Public ID key of the client, not required when using a linked custom domain (i.e., client.speedsize.com or cdn.website.com)

LINK
Absolute path to an image or video. Relative path can be used only in case of configured linked domain

PARAMS
Transformation parameters (see below)

PSYCHOVISUAL APPROACH TO COMPRESSION

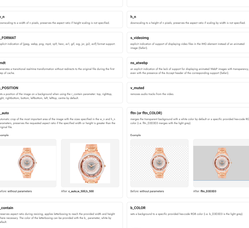
SpeedSize™ emulates the human brain and its visual perception during the process of compression. Knowing the acceptable level of compression is a key to a sweet spot of an image both being compressed to the minimal file size and maintaining the visual quality. The majority of compression techniques/tools don't do any quality assessment of the results they provide. But even those which do, usually use metrics such as SSIM, MSSIM, PSNR, MSE, etc that can't fully represent the real level of quality loss and were not designed to judge the level of compression in the first place.

Psychovisual AI Compression

SUPERIOR CDN PERFORMANCE

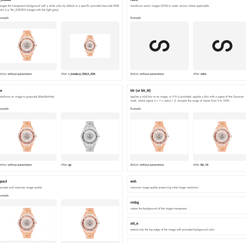
Once integrated to your website, on receiving an image or video first request, SpeedSize™ will:

1. SpeedSize™ identifies the user's browser type, service device, and screen DPR at the edge, adjusting the image or video resolution and format according to the provided client hints and accept-header of the cross attributes.
2. SpeedSize™ fetches the original media, temporarily delivering a real-time compressed or the original file to the user. In the meantime, it employs its proprietary Psychovisual AI compression to create the best media format using advanced resizing technology, maintaining image quality while minimizing file size.
3. Lastly, SpeedSize™ caches the AI-processed file, selecting the best size-to-quality ratio and the optimal media format for each individual image and video. It then delivers it from cache for all future users, guaranteeing the fastest media load times and an exceptional visual user experience.



600+ GLOBAL CDN LOCATIONS

AWS CloudFront's globally spread points of presence



TRANSFORMATION PARAMETERS

f_FORMAT
Forces a specific format to transfer JPEG, WEBP, PNG, APNG, GIFV, HEVC, AVC, SVF, SVG, JPG, JPE, AVIF, Tagged, JxRxx or File Size (in bytes) supported by the extension. Default value is the format of specified **f_path**, extension delivery of the best file available supported by the client, or the original file if the client hints header is not sent in the Accept-header and other parameters. If set, Cache is enabled by default.

Example

Before: **without parameters** After: **f_avif**

w_AUTO
Sets the width of the most important area of the image with the size specified in the **w** and **h** parameters, preserve the requested aspect ratio. If the specified width or height is greater than the original file.

Example

Before: **without parameters** After: **c_auto,w_300,h_300**

c_contain
preserve aspect ratio during resizing, applies letterboxing to each the provided width and height values (because). The color of the letterboxing can be provided with the **b_** parameter, only for **c_contain**

Example

Before: **without parameters** After: **c_contain,w_300,h_300**

c_inside
resizes the requested background with a white color by default or a specific provided Hex/web RGB color (i.e. Pink,200,0,0) image with the high gaps

Example

Before: **without parameters** After: **c_inside,w_300,h_300**

ign
transforms an image to grayscale (blackbrown)

Example

Before: **without parameters** After: **ign**

upscl
upscales and improves image quality

Example

Before: **without parameters** After: **upscl**

c_mannual
manual crop of the image with specified sizes in x, y and x, y and coordinates (x,y and x,y)

c_fill
center crop of all images of the image with the provided background color (b_COLOR, default black)

Chained transformations
Allows multiple transformation parameters using semicolon (;) to create a new image e.g. w_300,c_fill,b_red,f_png

mono_n
converting the monochrome (with b & y)pixels while increasing the aspect ratio to the nearest bigger resolution from the list of standard resolutions: (w: 16, 160, 190, 320, 360, 370, 390, 400, 480, 500, 640, 680, 800, 840, 960, 1024, 1080, 1280, 1366, 1440, 1600, 1680, 1920, 2048, 2560, 3840, 4096, 5120, 6400, 7680, 8192, 10240, 12800, 15360, 16384, 19200, 25600, 32768, 38400, 51200, 65536, 76800, 102400, 131072, 153600, 204800, 262144, 307200, 409600, 524288, 614400, 819200, 1048576, 1228800, 1638400, 2097152, 2621440, 3276800, 4194304, 5242880, 6553600, 8388608, 10485760, 13107200, 16777216, 20971520, 26214400, 33554432, 41943040, 52428800, 65536000, 83886080, 104857600, 131072000, 167772160, 209715200, 262144000, 335544320, 419430400, 524288000, 671088640, 838860800, 1048576000, 1342177280, 1677721600, 2147483647)

Example

Before: **without parameters** After: **mono_n,w_1024**

f_n
displaying the image in a single white increasing the aspect ratio to the nearest bigger resolution from the list of standard resolutions: (w: 16, 160, 190, 320, 360, 370, 390, 400, 480, 500, 640, 680, 800, 840, 960, 1024, 1080, 1280, 1366, 1440, 1600, 1680, 1920, 2048, 2560, 3840, 4096, 5120, 6400, 7680, 8192, 10240, 12800, 15360, 16384, 19200, 25600, 32768, 38400, 51200, 65536, 76800, 102400, 131072, 153600, 204800, 262144, 307200, 409600, 524288, 614400, 819200, 1048576, 1228800, 1638400, 2097152, 2621440, 3276800, 4194304, 5242880, 6553600, 8388608, 10485760, 13107200, 16777216, 20971520, 26214400, 33554432, 41943040, 52428800, 65536000, 83886080, 104857600, 131072000, 167772160, 209715200, 262144000, 335544320, 419430400, 524288000, 671088640, 838860800, 1048576000, 1342177280, 1677721600, 2147483647)

Example

Before: **without parameters** After: **f_n**

c_sidecolorg
explicit indication of support of displaying color-files in the MFD element instead of an animated image (GIFV)

c_sidecolorg
explicit indication of support of displaying color-files in the MFD element instead of an animated image (GIFV)

c_alwaysjpg
explicit indication of support of displaying color-files in the MFD element instead of an animated image (GIFV)

c_alwaysjpg
explicit indication of the lack of support for displaying uncoded PNGF image with transparency, and with the presence of the brutal header of the corresponding support (GIFV)

c_mnoted
converts static icons from the cross

f_AUTO
sets a background to a specific provided Hex/web RGB color (i.e. 0,200,000) or the high gaps

Example

Before: **without parameters** After: **f_AUTO**

f_COLOR
sets a background to a specific provided Hex/web RGB color (i.e. 0,200,000) or the high gaps

Example

Before: **without parameters** After: **f_COLOR**

f_COLOR
sets a background to a specific provided Hex/web RGB color (i.e. 0,200,000) or the high gaps

Example

Before: **without parameters** After: **f_COLOR**

bb (or bb_RC)
sets the image to the image, as if it is provided, applies letter-with a space of the Discover cross (or signs (x,y,x,y) (i.e. 0,0,0,0). If included the range of values from 0 to 9999.

Example

Before: **without parameters** After: **bb,10**

auto
auto-detects the image, and the best quality and format are chosen for the image with provided background color

c_fill_b
center crop of all images of the image with the provided background color

c_fill_r
center crop of all images of the image with the provided background color

c_fill_t
center crop of all images of the image with the provided background color

c_fill_l
center crop of all images of the image with the provided background color

c_fill_n
center crop of all images of the image with the provided background color

Custom Parameters
SpeedSize™ offers a variety of customizable transformation parameters tailored to client preferences

ADAPTIVE RESIZING

Many websites suffer from poor loading times and user experiences due to inaccurate resizing parameters. This issue affects images and videos across various pages, which are delivered at an overly high resolution and then downscaled by the browser.

In the example below, you can see that "Rendered size" (the actual size of the images was rendered by the browser) is 1024x577, "Intrinsic size" (the resolution of the requested image) is 3024x1280

The current file size of the image (excluded for the time being) should be only 70.5 KB

SpeedSize™ CDN provides multi-tiered auto-image resizing. By default, it detects the user's device type and serves suitable image sizes based on mobile, tablet, and desktop resolutions. However, for more precision (even pixel-perfect resizing), you can achieve further adjustments using one or a combination of the following:

1. As a best practice, ensure that the smallest width you specify in srcset is at least as wide as the widest space the image can occupy on your site. This will ensure that AI-optimized images and videos always look crisp and clear and true to the original image or video.
2. SpeedSize™ CDN utilizes Client Hints and the sizes attribute for exact image adjustments. Client Hints are HTTP headers that inform about the user's display, desired image resolution, and connection. The browser calculates expected image resolution using the sizes attribute per image. These features can be easily activated in server settings.
3. SpeedSize™ CDN supports a URL rewriting which can be used for resizing and other transformations based on the unique URL patterns of images and videos. Example: https://website.com/path/image.jpg. This may be configured to be translated https://website.com/target/image.jpg or to s3://bp-width, or https://website.com/-/AuroraDirect/image.jpg to s3://bp-width etc.
4. Accept-CH = "Sec-ch-Width, Width, DPR" Feature Policy = "ch-viewport-width https://ch.speedsize.com, ch-dpr https://ch.speedsize.com, ch-width https://ch.speedsize.com"

WORKING FLOW WITHOUT SPEEDSIZE CDN

SpeedSize™ CDN guarantees the surfeiest Time To First Byte (TTFB) loading speed for images and videos in the market. Achieve this by seamlessly integrating a straightforward prefix into your media URLs. If you are unable to utilize AWS CloudFront's points of presence and caching hardware infrastructure and prefer not to employ SpeedSize's patented software technologies for ensuring the fastest and most stable performance for your website's media, you can still follow the instructions below. Perform a custom integration exclusively between SpeedSize AI compression and your preferred CDN.

1. SpeedSize updates the cache upon progress of processing. Once the final processing is done, the Cache-Control response header value is set to max-age=31536000 (1 year).
2. Custom CDN replicates its cache based on CloudFront's Cache Control response header and clients Accept request header values. Once the Vary response header exceeds the value of max-age of Cache-Control response header, the cache is invalidated (the trust CDN happens automatically).
3. Clients communicate with custom CDN and receive media content generated by SpeedSize exclusively.
4. Custom CDN requests CloudFront origin on cache-miss events passing the Accept request header from the client.

If you have any questions, need us to provide you a **Client_ID** or technical integration support needed, please contact us: support@speedsize.com
This document is confidential to SpeedSize™ and is for technical review only. It should not be disclosed or used for any other purpose.