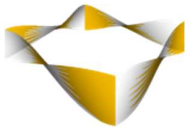


JaJuMa

WebP Optimized Images

10/2020

Table of Contents

1. Introduction

Performance is getting increasingly important when it comes to User Experience and also SEO. One way to improve page load times is reducing the amount of data that needs to be transferred. Hence, with images usually making up for the biggest share of data to be downloaded for a webpage, it is crucial to choose the best file format and compression to reduce image sizes as much as possible.

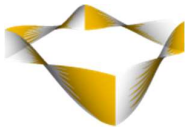
Magento by default supports .jpg + .png images as file formats. Each of these can be optimized by applying lossless as well lossy compression.

However, with newer image file format WebP, size of images can be reduced even more with its superior compression and quality characteristics compared to their older JPEG and PNG counterparts.

JaJuMa WebP Optimized Images Extension adds WebP Support to new as well as existing Magento Stores. There is no need to change existing Product Images. Also new images can be uploaded as usual as JPG or PNG, the Extension will create and handle WebP variants automatically.

For further details as well as comparisons between .jpg /.png and .webp regarding file size and quality, please see:

https://www.jajuma.de/en/jajuma-develop/extensions/webp-optimized-images-extension-for-magento-2



2. System Requirements & Installation

For the extension to work correctly, at least one of the supported conversion tools must be available on your server. See section → 2.2 - Conversion Tools for further details.

For installing the extension, follow installation process as with any Magento Extension.

2.1. Manual Extension Installation via FTP

For manual installation by FTP, please follow these steps:

Before Installing

1. We recommend you to duplicate your live store on a staging/test site and try installation on your staging/test site before deploying to your live store
2. Backup Magento files and the store database

Please Note: It's very important to backup all themes and extensions in Magento before installation, especially when you are working on a live server.
We strongly recommend you to do not skip this step.

Upload the Extension

1. Log into your hosting space via a FTP client (e. g. FileZilla, WinSCP, cuteFtp)
2. Create Folder: <magentoroot>/app/code/Jajuma/WebpImages
3. Unzip extension package and upload files into: <magentoroot>/app/code/Jajuma/WebpImages
4. Enter and run the following commands at the command line:

```
php bin/magento setup:upgrade
php bin/magento setup:static-content:deploy
```

2.2. Conversion Tools

This Extension supports 3 different conversion tools:

1. GD
2. cwebp +
3. Imagemagick

At least one of these conversion tools must be available on your server.
GD should work out of-the-box on any Magento installation.
But the command line tools cwebp + Imagemagick might need additional installation or configuration steps to run correctly on your server.

In case of any issues with these tools or in case you are unsure how to get these tools up and running, please ask your hosting provider for assistance.

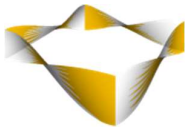
2.2.1. Check Conversion Tools

Check from Backend

If you have the extension already installed you can check if the conversion tools are working on your server using the Conversion Test Tool from Backend.
→ 4.2.4 - Test & Preview Conversion

Check from CLI

If you want to check which conversion tools are available on your server from CLI / without having the extension installed, please follow these steps:



2.2.1.1. Check GD from CLI

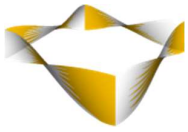
Run following command on your CLI:

```
php -i | grep -E 'WebP Support|GD'
```

If GD is
- not supported / installed on your server: This will give you an empty result.
- supported / installed on your server: Output should be something like this:

```
GD Support => enabled
GD Version => bundled (2.1.0 compatible)
WebP Support => enabled
```

Note: The last line of this output must be showing enabled.



2.2.1.2. Check cwebp from CLI

Run following command on your CLI see if cwebp is available as global command:

```
cwebp
```

If cwebp is
- not supported / installed on your server: This will give you an you an error message.
- supported / installed on your server: Output should be the man page for cwebp.

In case cwebp is not available as global command on your server, you can try these steps:

- Add cwebp to you server

a) Upload official cwebp lib from here https://developers.google.com/speed/webp/docs/precompiled
b) This lib is also bundled with our extension.
If you have the extension already installed you will find it in this path:
/magento/root/app/code/Jajuma/WebpImages/bin/cwebp

- Make sure the lib is executable

- Run the following command on your CLI:

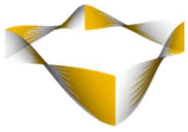
```
</path/to/cwebp/lib/on/your_server>/cwebp
```

If cwebp
- can run on your server using this method: Output should be the man page for cwebp.
- cannot run on your server using this method: This will give you an error message / note about missing dependencies.

Please Note: For cwebp to work correctly with the extension:

➔ cwebp and all dependencies need to be installed correctly on your system
➔ cwebp binary needs to be executable (by webuser)

If needed, please check with your hosting provider to get cwepb up and running and to ensure these requirements are met.



2.2.1.3. Check Imagemagick from CLI

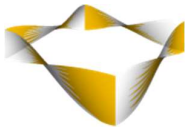
Run following command on your CLI to see if ImageMagick is available as global command:

```
convert
```

If Imagemagick is
- not supported / installed on your server: This will give you an error message.
- supported / installed on your server: Output should be the man page for Imagemagick.

If Imagemagick is available from any other custom command.

If needed, please check with your hosting provider to get Imagemagick up and running.



3. Extension Scope

JaJuMa WebP Optimized Images Extension supports conversion of .jpg and .png images into .webp files based on configuration as described below using one of three supported conversion tools:

- GD
- Cwebp
- Imagemagick

All images found in markup on all pages on your site, as well as images in Fotorama Media Gallery on Product Page are converted and delivered as WebP images to visitors.
Using HTML5 <picture>-tag / browser WebP support detection it is ensured that .webp files are only served to browsers that can show this image type.
For visitors using a web browser without .webp support, the extension automatically fall back to delivering .jpg or .png files.

Additionally, the extension supports native lazy loading by optionally adding the *loading="lazy"* tag to converted images.

The extension integrates with Magento Media Management.
Means, WebP image files are created for the same size variants as for .jpg/.png files (for example thumbnails and small grid images as well as original size).
The extension comes with a separate Media Cache for WebP images.
WebP images generated and stored in this Cache can be cleared at any time,
e. g. to apply new conversion config / quality level.
JaJuMa WebP Optimized Images Extension is fully FPC (Full Page Cache) compatible.

As any Extension on Magento Marketplace, JaJuMa WebP Optimized Images is developed to work with Magento Default. While trying as far as feasible to avoid conflicts with, or breaking other custom functions, we cannot guarantee so for every imaginable customization out there.
Specifically, in case you have other customizations regarding how images are delivered to your customers (such as lazy loading, image swap on hover, images served from CDN...), the extension might still work fine, but it is also possible there might be conflicts preventing the extension from working as expected.
It totally depends on how your custom implementation is designed to work.

However, besides supporting 3 different conversion tools we also added some configuration options trying to have the extension work in as many scenarios as possible (See → 4.4 - Advanced Configuration).

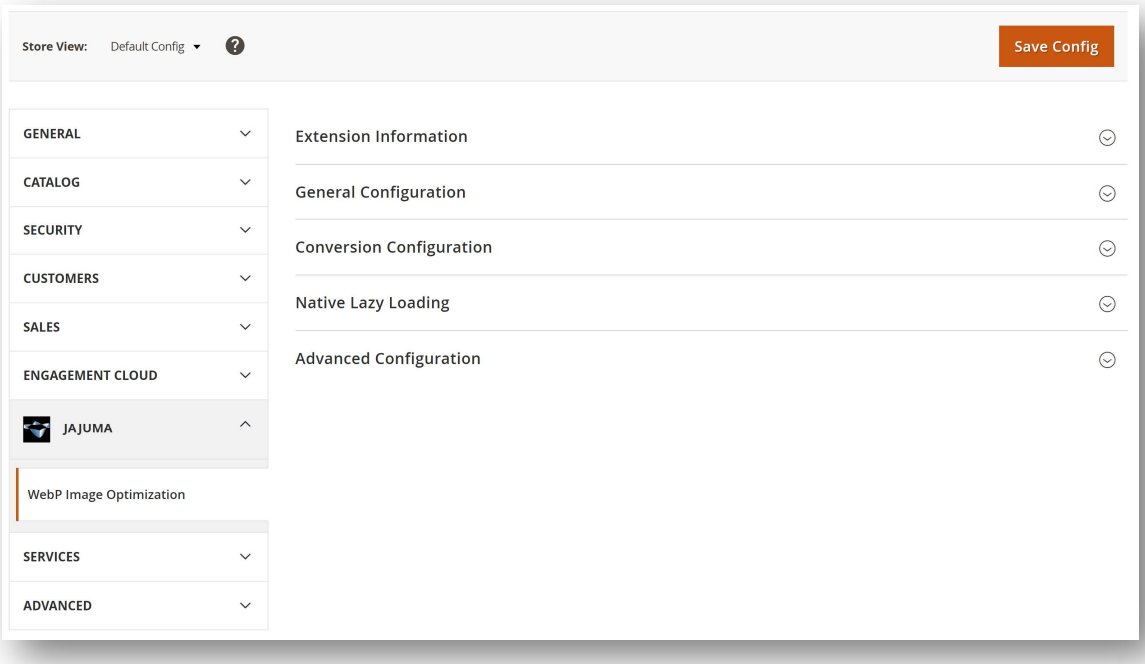
4. Configuration

In Magento Backend see

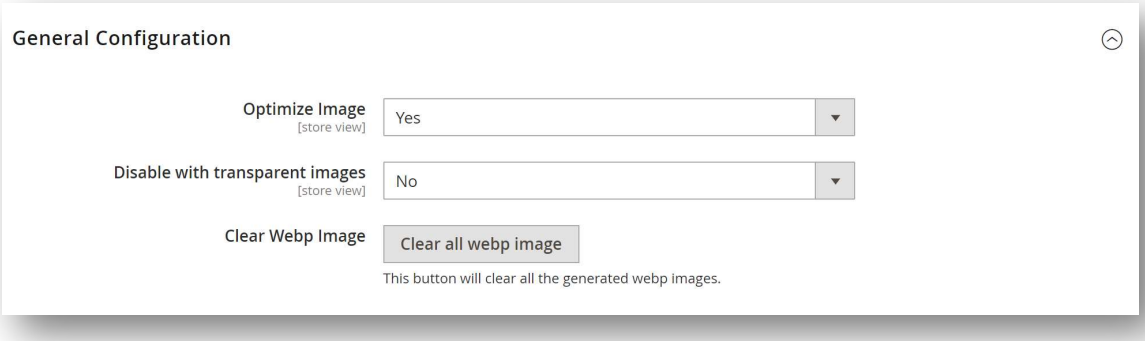
Stores -> Configuration -> Open Tab: JaJuMa -> Select: WebP Image Optimization

For JaJuMa WebP Optimized Images configuration with 3 sections:

- General Configuration
- Conversion Configuration
- Native Lazy Loading
- Advanced Configuration



4.1. General Configurations



4.1.1. Optimize Images

To enable WebP Optimized Images, select from Drop Down:

- Yes → Enable Extension
- No → Disable Extension

If enabled, the extension will convert existing images to WebP as configured (see → 4.2 - - 13 -Conversion Configurations) and deliver those to Store visitors.

4.1.2. Disable with transparent images

In case you see increased file sizes or quality issues with images that have transparency, you can disable WebP to be used for these images by selecting from Drop Down:

- Yes → Images with transparency will be excluded
- No → Images with transparency will be included

However, in most cases it should be fine to keep this as “No”.

4.1.3. Clear WebP Image Cache

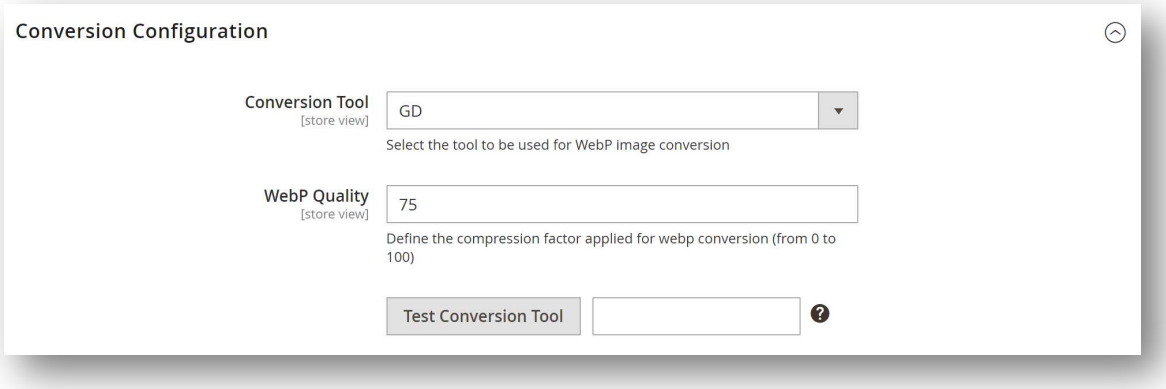
Click this button to clear all previously created WebP images.
E. g. in case you want to change conversion settings to have WebP images with higher or lower quality.
New WebP images will be created automatically on the fly with following page requests.

!!! Please Note !!!
When flushing WebP image Cache, please do not forget to also flush your Full Page Cache if applicable.

4.2. Conversion Configurations

Please see below for configurations regarding conversion depending on conversion tool you want to use.
For the conversion to WebP to work, you need to make sure the conversion tool selected is setup and installed correctly on your server.
See below and section → 2.2 - Conversion Tools for further details.
Please Note: For testing different conversion settings you may use Conversion Test Tool from Backend. → 4.2.4 - Test & Preview Conversion
For changing conversion tool / parameters also for previously created WebP images, just clear WebP Image Cache. This will delete generated WebP files.
By this, new image files will be created applying your new conversion configuration.
→ 4.1.3 - Clear WebP Image Cache

4.2.1. Config Image Conversion Using GD



For using GD as conversion tool, you just need to

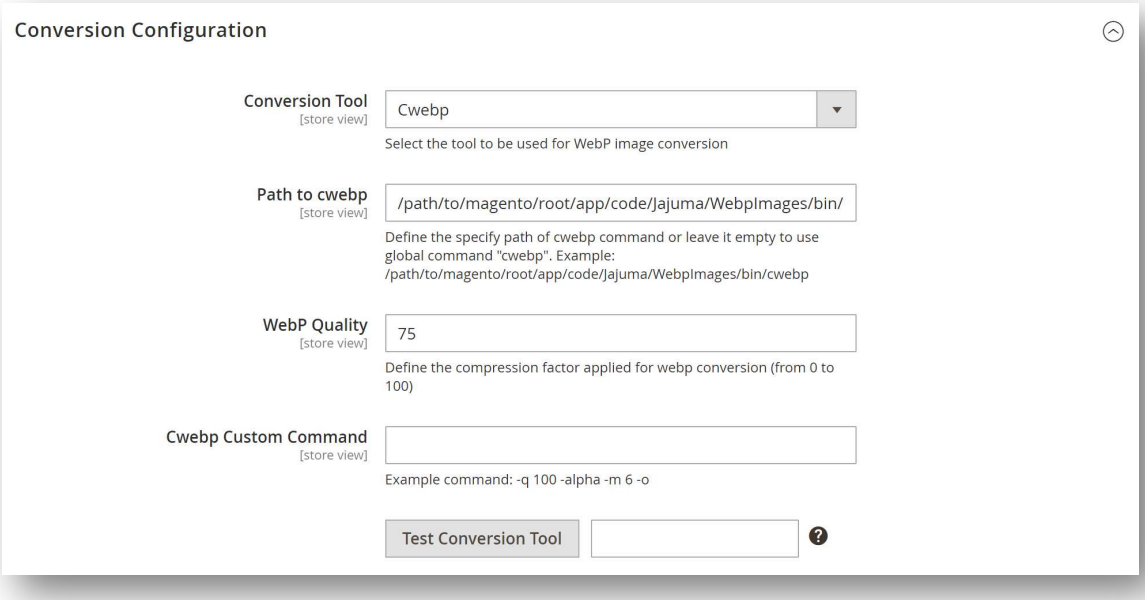
1. **Choose "GD" from Conversion Tool Drop-Down**

2. **WebP Quality:** Define the compression factor to be applied**.**
   (GD does not provide any further conversion options.)

3. **Test WebP Conversion using GD**

   → See 4.2.4 - Test & Preview Conversion

4.2.2. Config Image Conversion Using cwebp



For using “Cwebp” as conversion tool, you just need to

1. **Choose “Cwebp” from Conversion Tool Drop-Down**
2. **Path to cwebp**

   Configure correct path to working cwebp executable.

   Please note: For cwebp to work correctly

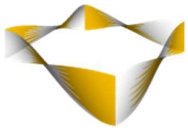
   ➔ cwebp and all dependencies need to be installed correctly on your system
   ➔ cwebp binary needs to be executable (by webuser)

   Option 1

   The cwebp binary is included with the Extension.
   In order to use the bundled binary please use following path:

   [magentoroot]/app/code/Jajuma/WebpImages/bin/cwebp

   Please double-check the binary is executable by your web-user and make sure all required dependencies are installed on your system.



Option 2

You can also use any cwebp installation already present on your system or download from https://developers.google.com/speed/webp/download and install in any other path.

Just add the path to the executable binary.

Option 3

In case cwebp is executable via global command as “cwebp”,
you can just leave this configuration empty.

4. **Conversion Parameters:**

Base Mode

WebP Quality:
Define the compression factor to be applied.

Conversion will be performed using following pre-configured command:

```
-q <WebP Quality Config> -alpha_q 100 -z 9 -m 6 -segments 4 -sns 80 -f 25 -sharpness 0 -strong -pass 10 -mt -alpha_method 1 -alpha_filter fast
```

Advanced Mode

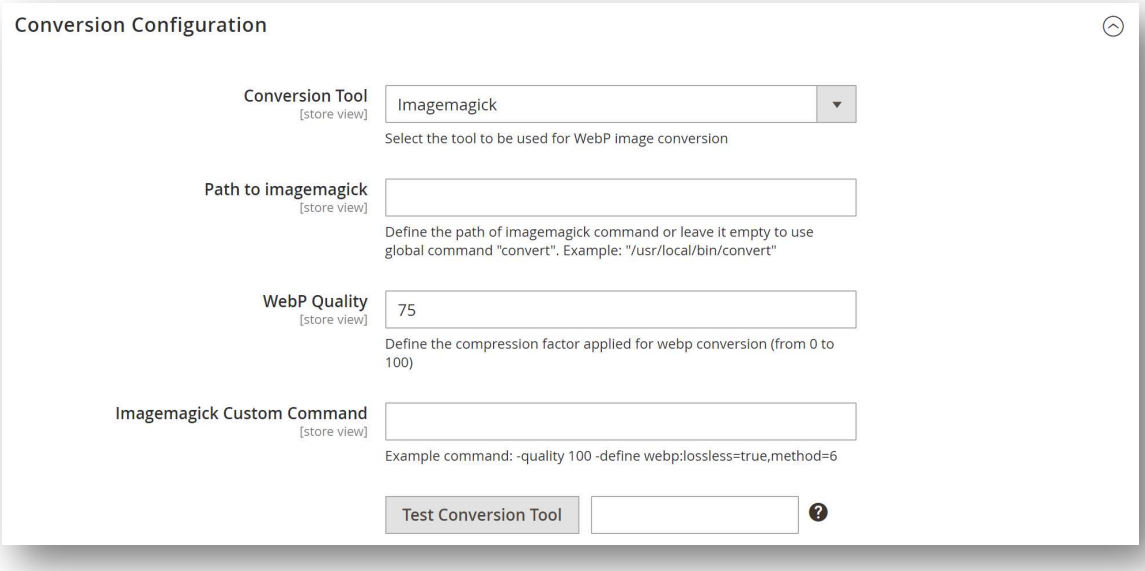
Cwebp Custom Command:
Define any custom cwebp command with parameters that suit your needs.
For a detailed description of parameters available, please see
https://developers.google.com/speed/webp/docs/cwebp

Note: Leave this configuration empty to use Base Mode

5. **Test WebP Conversion using cwebp**

→ See 4.2.4 - Test & Preview Conversion

4.2.3. Config Image Conversion Using Imagemagick



For using “Imagemagick” as conversion tool, you just need to

1. **Choose “Imagemagick” from Conversion Tool Drop-Down**

2. **Configure correct path to Imagemagick executable**

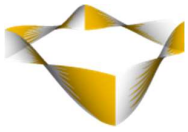
   Option 1

   In case Imagemagick is available through global command "*convert*", you can just leave this configuration empty.

   Option 2

   Or define the custom path of imagemagick command on your system.

   Example: */usr/local/bin/convert*



3. **Conversion Parameters:**

Base Mode

WebP Quality:
Define the compression factor to be applied.
Conversion will be performed using following pre-configured command:

```
-quality <WebP Quality Config> -define
webp:lossless=false,method=6,segments=4,sns-strength=80,auto-
filter=true,filter-sharpness=0,filter-strength=25,filter-type=1,alpha-
compression=1,alpha-filtering=fast,alpha-quality=100
```

Advanced Mode

Imagemagick Custom Command:
Define any custom Imagemagick command with parameters that suit your needs
For a detailed description of parameters available, please see https://www.imagemagick.org/script/webp.php

Note: Leave this configuration empty to use Base Mode

4. **Test WebP Conversion using Imagemagick**

→ See 4.2.4 - Test & Preview Conversion

4.2.4. Test & Preview Conversion

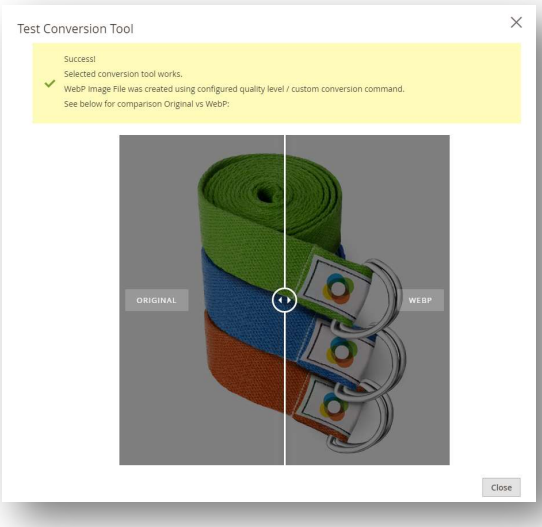
For testing the conversion and preview the conversion result please follow these steps:

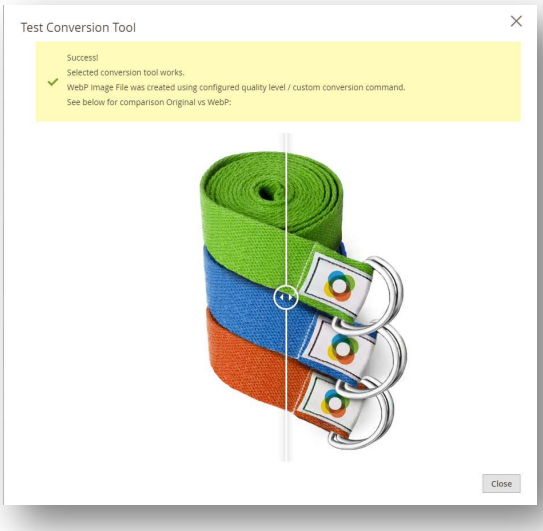
1. Complete Conversion Configuration as described in section 4.2
2. Input the Product ID (not SKU) into the text field next to "**Test Conversion Tool**" - Button

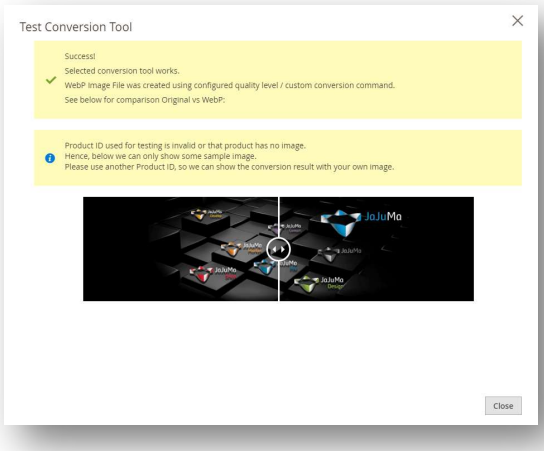
3. Click "**Test Conversion Tool**" - Button
4. See PopUp with conversion result:

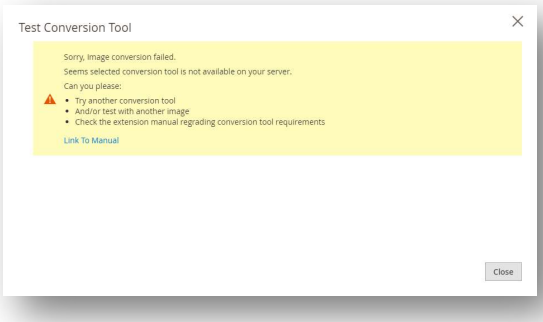
4.1. **Success:** In case the conversion is successful:
You will see the result as a side by side comparison between original image (left) and created WebP image (right).
You can move the vertical slider to reveal more of the original image or the WebP image as you like, to get the perfect impression regarding quality.

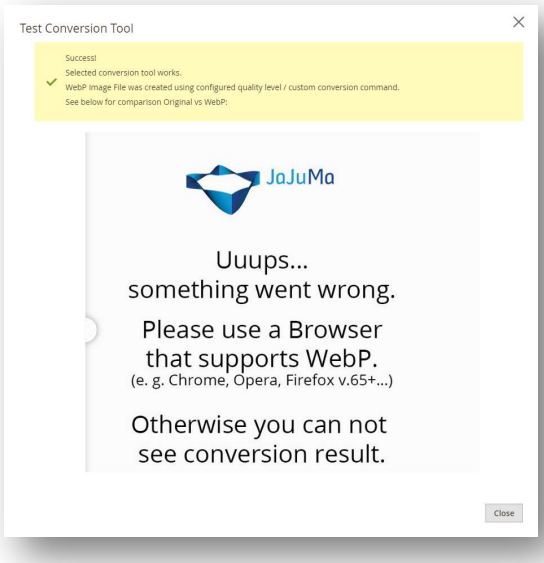
4.2. **Fail:** In case the conversion fails the PopUp will give you some hint on what went wrong and how to fix it.



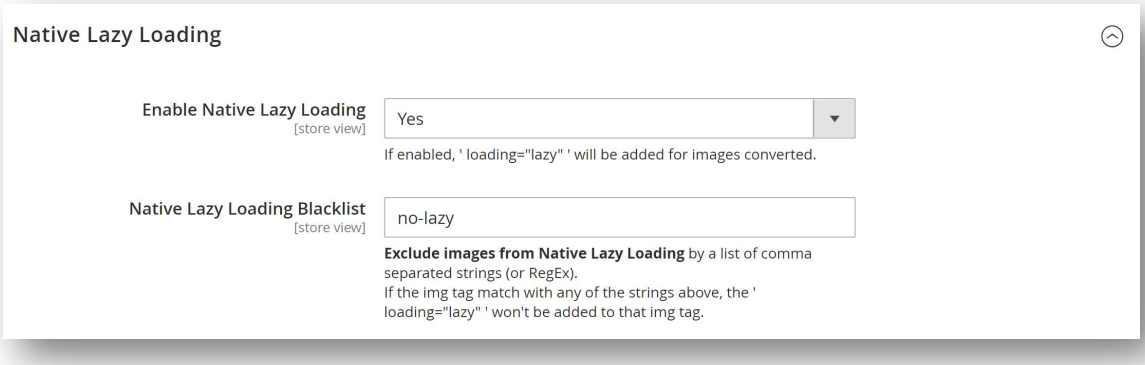



Test Conversion Tool
Success!
Selected conversion tool works.
WebP image File was created using configured quality level / custom conversion command.
See below for comparison Original vs WebP:
Product ID used for testing is invalid or that product has no image.
Hence, below we can only show some sample image.
Please use another Product ID, so we can show the conversion result with your own image.
Close


Test Conversion Tool
Sorry, image conversion failed.
Seems selected conversion tool is not available on your server.
Can you please:
Try another conversion tool
And/or test with another image
Check the extension manual regrading conversion tool requirements
Link To Manual
Close


Test Conversion Tool
Success!
Selected conversion tool works.
WebP image File was created using configured quality level / custom conversion command.
See below for comparison Original vs WebP:
JaJuMa
Uuups...
something went wrong.
Please use a Browser
that supports WebP.
(e. g. Chrome, Opera, Firefox v.65+...)
Otherwise you can not
see conversion result.
Close

4.3. Native Lazy Loading



To enable Native Lazy Loading, select from Drop Down:

- Yes → Enable Native Lazy Loading
- No → Disable Native Lazy Loading

If enabled, the extension will add the *loading="lazy"* tag to converted images.

4.3.1. Native Lazy Loading Blacklist

With Native Lazy Loading Blacklist config it is possible to exclude certain images from native lazy loading. This is recommended e. g. for above-the-fold images.
You can add any string or RegEx as comma separated list.
Any <img>-tag matching with any of the given strings/RegEx will be excluded from conversion.

Example 1: Exclude Logo named *logo.png*

```
logo\.jpg
```

Example 2: Exclude Logo named *logo.png* AND all images through WYSIWYG editor

```
logo\.jpg,wysiwyg
```

Example 3: Exclude images with a certain css class, e. g. banner-image

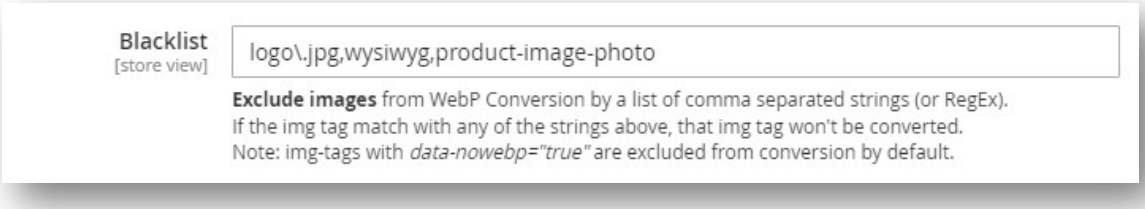
banner-image

4.4. Advanced Configuration

Note:

For most sites following configs are not needed.
We recommend to use only if you understand what you are doing.

4.4.1. Blacklist



With Blacklist config it is possible to exclude certain images on your site from WebP conversion. You can add any string or RegEx as comma separated list.
Any <img>-tag matching with any of the given strings/RegEx will be excluded from conversion.

Example 1: Exclude Logo named *logo.png*

logo\.jpg

Example 2: Exclude Logo named *logo.png* AND all images through WYSIWYG editor

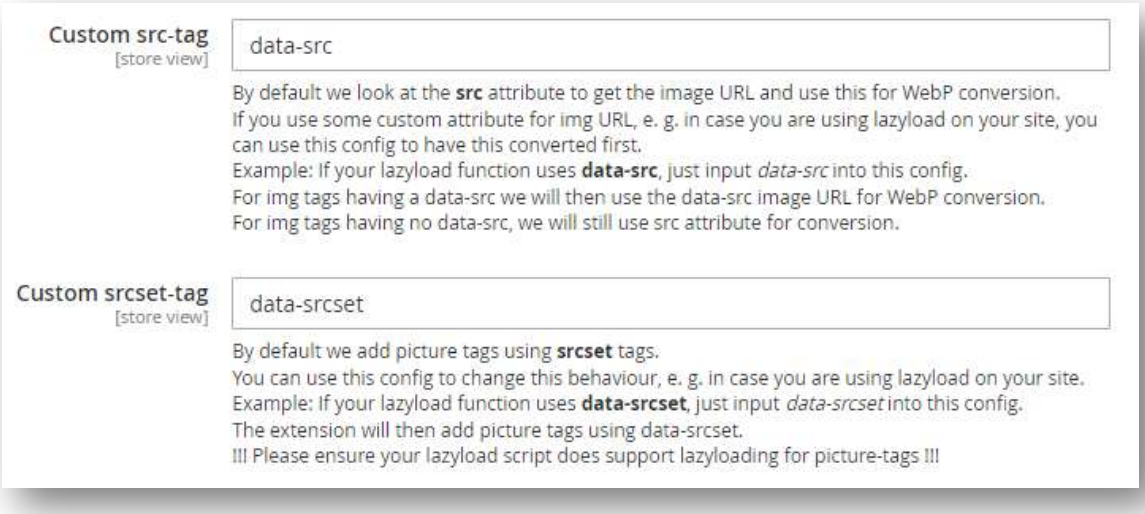
logo\.jpg,wysiwyg

Example 3: Exclude images with a certain css class, e. g. product-image-photo

product-image-photo

Note: img-tags with data-nowebp="true" and images in media/captcha are excluded from conversion by default.

4.4.2. Custom src-/srcset-tags



With this config it is possible to make this extension work with some custom cases where images are loaded via some lazyloading function.

By default we look at the **src** attribute to get the image URL and use this for WebP conversion.
If you use some custom attribute for img URL, e. g. in case you are using lazyload on your site, you can use this config to have this converted first.
Example: If your lazyload function uses **data-src**, just input *data-src* into this config.
For img tags having a data-src we will then use the data-src image URL for WebP conversion.
For img tags having no data-src, we will still use src attribute for conversion.

By default we add picture tags using **srcset** tags.
You can use this config to change this behaviour, e. g. in case you are using lazyload on your site.
Example: If your lazyload function uses **data-srcset**, just input data-srcset into this config.
The extension will then add picture tags using data-srcset.

!!!
Note:
Please ensure your lazy load script does support lazy loading for picture-tags
!!!

5. CSS Background Images

Please note: CSS background images are not converted automatically.
However, JaJuMa WebP Optimized Images extension enables you to use .webp images as CSS backgrounds on your site easily.

In browsers that do not support .webp, our extension will automatically add a CSS class *".no-webp"* to the <*body*> tag, e. g. like this:

```
<body class="cms-home cms-index-index page-layout-1column no-webp">
```

This enables you to use browser compatible .webp images as backgrounds.
If you have a background image added by CSS on your site like this:

```
.your .css .class {
    background: url(../path/to/background.png);
}
```

You can simply change it to this to use your manually converted .webp image as background and keep your .png/.jpg as fallback for old browsers:

```
.your .css .class {
    background: url(../path/to/background.webp);
}
body.no-webp .your .css .class {
    background: url(../path/to/background.png);
}
```

6. Further Recommendations

6.1. General

Conversion results, in terms of webp file size, saving compared to .jpg/.png as well as quality, highly depend on your existing images. Hence it is not possible to give a "one size fits all" recommendation or provide a configuration that delivers 'perfect' results for every use case.

With the preconfigured conversion parameters in Base Mode we tried our best to provide you with a configuration that provides good results out-of-the-box.
In case the results with this configuration are not satisfying to you, please feel free to play around with the different parameters in Advanced Mode to find a configuration that better suits your needs. Its as easy as saving your custom command and deleting Magento Media Cache to see the results.

6.2. Which Tool Should I Use?

We recommend to use cwebp, the official conversion tool developed by Google.
It is the best choice regarding quality, conversion speed and also provides the most conversion parameters.
However, the price for this is the "challenge" to get cwebp up and running on your server, which might not be possible in every case, e. g. with some shared hosts.

In that case we recommend to use GD, it should work out-of-the-box on any Magento System without further setup requirements and is easy to use due to having just one option to be configured. But in most cases GD still provides pretty good results.

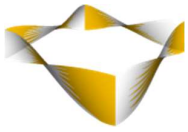
Imagemagick is also pretty good regarding conversion speed, however it is said to produce slightly less good quality than the other 2 tools.

6.3. What "WebP Quality" Should I Use?

This value defines the compression factor applied for conversion - Default is 75.

In case of lossy compression (default), a small value produces a smaller file with lower quality. Best quality is achieved by using a value of 100.

In case of lossless compression (specified by the -lossless option), a small factor enables faster compression speed, but produces a larger file. Maximum compression is achieved by using a value of 100.



6.4. How About All These Other Paramters?

Please see Official Documentation for

cwebp: https://developers.google.com/speed/webp/docs/cwebp

Imagemagick: https://www.imagemagick.org/script/webp.php

7. Support

Please feel free to contact JaJuMa support team via support@jajuma.de.
In case any additional information is required. We'd be more than happy to assist in setting up the extension.