

WebP Generator Module



version 1.0.4
de PrestaChamps

Configuration requise	2
Vue d'ensemble du module	3
Configuration du module	4
EWWW Cloud Convert	6
Paramètres de conversion CWEBP	7
Régénérer WebP	8
Prise en charge du navigateur WEBP	10
Compatibilité Prestashop 1.6	11

Configuration requise

Assurez-vous que votre version PHP est au moins de 5.6 sinon le module ne fonctionnera pas (comme indiqué dans la fiche dedu produit dans Addons PrestaShop)

configuration Sur le serveur, l'un des prochains modules PHP doit être configuré:

Cwebp => fonction d'exécution php activée

Imagick => module Imagick php avec webp compilé

Gmagick => module php Gmagick avec webp compilé

Gd => Gd module php avec les fonctions imagewebp, imagecreatefrompng et imagecreatefromjpeg

Ewww => ewww.io abonnement et service php curl

Aperçu du

module Ce module est un élément essentiel si vous souhaitez un site Web rapide.

Le module WebP Generator, créé pour les sites Prestashop, permettra aux utilisateurs de régénérer efficacement les images de leurs produits, catégories et fabricants, augmentant ainsi la vitesse du site.

Cette nouvelle technologie offre une compression sans perte et avec perte supérieure pour les images de votre site Web. Une fois que vous aurez commencé à générer des images via notre module, cela libérera de l'espace sur votre serveur. De plus, la qualité de vos images ne sera pas affectée.

Les images générées via le **module générateur WebP** sont 26% plus petites que les images PNG. Les images avec perte WebP sont 25 à 34% plus petites que les images JPEG comparables à indice de qualité SSIM équivalent.

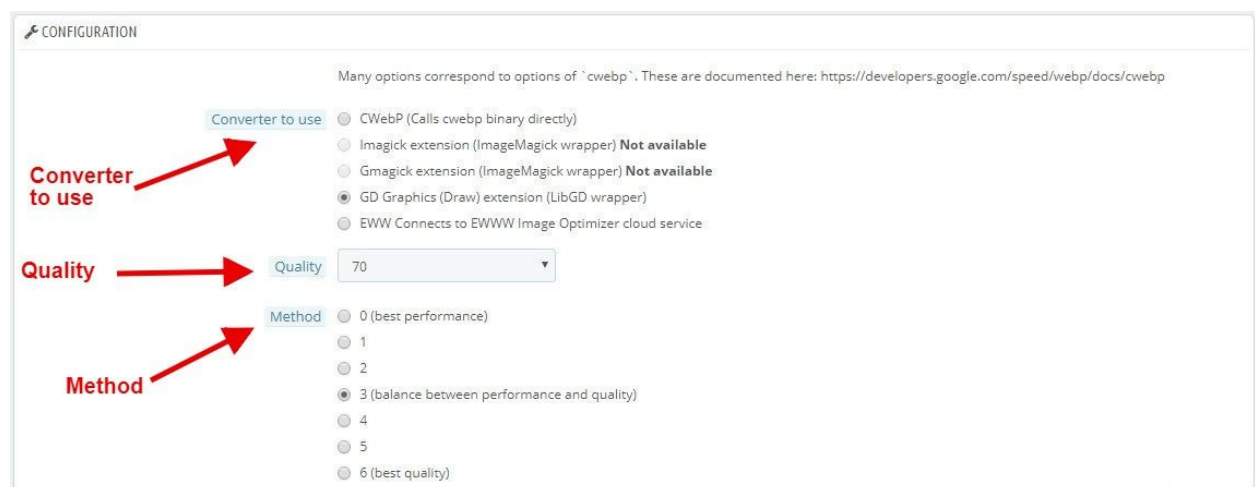
Info: - le module peut être installé à partir du back office du site web, comme tous les autres modules.

Configuration du module

- Une fois l'installation terminée, vous pouvez accéder au module en cliquant sur le **Configurer** bouton.



- **Convertisseur à utiliser** - Définissez les méthodes de conversion à utiliser en fonction de la configuration de votre serveur.
- **Qualité** - Spécifiez le facteur de compression pour les canaux RVB compris entre 0 et 100.
- **Méthode** - Ce paramètre contrôle le compromis entre vitesse de codage, taille et qualité du fichier compressé. Les valeurs possibles vont de 0 à 6. 0 est le plus rapide. 6 résultats dans la meilleure qualité.



- **Mémoire faible:** réduisez l'utilisation de la mémoire par un codage avec perte au coût d'environ 30% de temps de codage plus long et d'une taille de sortie légèrement plus grande.
- **perte** - SansEncode l'image sans aucune perte. L'option est ignorée pour lesPNG (forcé vrai)
- **métadonnées-** Métadonnées à copier de l'entrée à la sortie, le cas échéant.

Low memory


Reduce memory usage of lossy encoding by saving four times the compressed size (typically). This will make the encoding slower and the output slightly different in size and distortion. This flag is only effective for methods 3 and up, and is off by default. Note that leaving this flag off will have some side effects on the bitstream: it forces certain bitstream features like number of partitions (forced to 1)

Lossless

Recommended to no

Metadata ☐ All
☒ None (recommended)
☐ EXIF
☐ ICC
☐ XMP

Note: Only cwebp supports all values. gd will always remove all metadata. ewww, imagick and gmagick can either strip all, or keep all (they will keep all, unless metadata is set to none)

 Save

- **Mode démo** - Activez le «mode démo» pendant le processus de génération d'image. Ceci est recommandé pour éviter les erreurs 404 d'images non encore générées. Une fois le processus terminé, vous pouvez désactiver cette option.

Demo mode

This option is recommended during image generation process to avoid 404 errors of images that have not yet been generated. Once the process is complete you can turn off this option.

 Save

EWWW Cloud Convert

L'optimiseur d'images EWWW est un service de cloud computing très économique pour l'optimisation des images. Après l'achat d'une clé API, ajoutez le convertisseur dans l'option extra-convertisseurs, avec le champ de clé contenant la clé API.

L'API EWWW ne prend pas en charge l'option sans perte, mais convertit automatiquement les fichiers PNG sans perte. Les métadonnées sont soit tout ou rien. Si vous avez défini une autre option, toutes les métadonnées seront préservées.

EWWW CLOUD CONVERT

EWWW Image Optimizer is a very cheap cloud service for optimizing images. After purchasing an API key, add the converter in the extra-converters option, with key set to the key.

The EWWW api doesn't support the lossless option, but it does automatically convert PNG's losslessly. Metadata is either all or none. If you have set it to something else than one of these, all metadata will be preserved.

API Key

API Key

Save

Paramètres de conversion CWEBP

- **Utiliser la commande `nice`** - Si la commande `nice` est trouvée sur l'hôte, le binaire est exécuté avec une priorité faible afin de préserver les ressources système.
- **Essayez les chemins système communs** - Cette option teste si cwebp est disponible dans un chemin système commun (par exemple, /usr/bin/cwebp..)
- **Essayez le binaire fourni** - Si CWebP n'est pas installé sur le serveur, le fichier binaire fourni est sélectionné dans Convertisseurs / Fichiers binaires (selon le système d'exploitation) - après la validation de la somme de contrôle
- **Active le filtrage automatique** - Cet algorithme passera temps supplémentaire optimisant la force de filtrage pour atteindre une qualité bien équilibrée. Malheureusement, cela coûte extrêmement cher en termes de calcul. Il faut environ 5 à 10 fois plus de temps pour effectuer une conversion. Une image de 1 Mo dont la conversion prend environ 2 secondes en moyenne prend environ 15 secondes pour la conversion avec filtrage automatique. Donc, dans la plupart des cas, vous voudrez laisser cette valeur par défaut, qui est désactivée.
- **Options de ligne de commande** - Cela vous permet de définir tout paramètre disponible pour cwebp de la même manière que vous le feriez lors de l'exécution de cwebp. Vous pouvez par exemple le définir sur "-sharpness 5 -mt -crop 10 10 40"

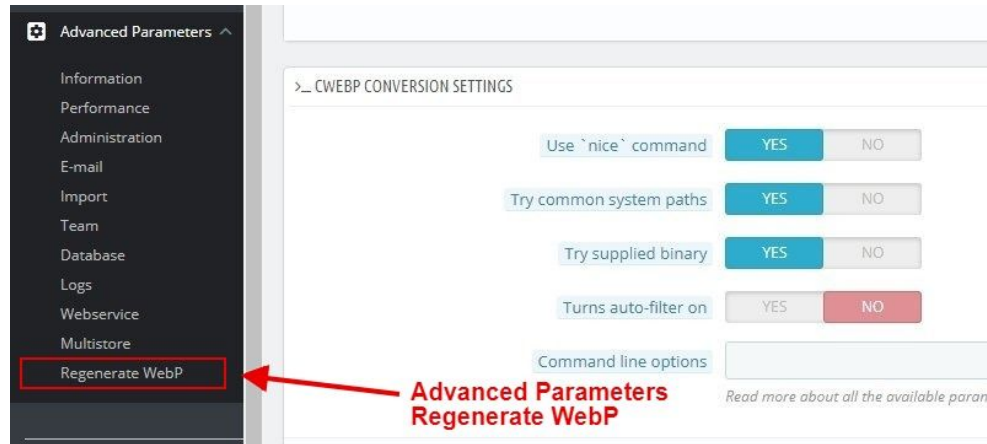
The screenshot shows a web interface titled ">_ CWEBP CONVERSION SETTINGS". It contains five toggle switches and a text input field, each with a red arrow pointing to it from the right. The toggles are: "Use `nice` command" (YES selected), "Try common system paths" (YES selected), "Try supplied binary" (YES selected), and "Turns auto-filter on" (NO selected). The text input field is labeled "Command line options" and has a red arrow pointing to it from the left. Below the input field is a link: "Read more about all the available parameters here: https://developers.google.com/speed/webp/docs/cwebp#additional_options". At the bottom right, there is a "Save" button with a floppy disk icon.

En savoir plus sur tous les paramètres disponibles ici:

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

Regenerate WebP

Une fois que vous avez configuré le module, veuillez vous reporter à «**Paramètres avancés**» dans le menu de gauche. Choisissez l' «**Regenerate WebP**» onglet.



Vous pouvez choisir les images que vous souhaitez générer.

- Images du produit
- Catégorie Images Images du
- fournisseur Images du
- magasin Images du
- fabricant

Une fois que vous avez choisi, vous pouvez commencer à générer. Vous pouvez voir le nombre d'images pour chaque type et la barre de traitement affiche le pourcentage des images générées.

? WebP is a modern image format that provides superior **lossless and lossy** compression for images on the web. Using WebP, webmasters and web developers can create smaller, richer images that make the web faster.

WebP lossless images are 26% smaller in size compared to PNGs. WebP lossy images are 25-34% smaller than comparable JPEG images at equivalent SSIM quality index.

Lossless WebP **supports transparency** (also known as alpha channel) at a cost of just 22% additional bytes. For cases when lossy RGB compression is acceptable, **lossy WebP also supports transparency**, typically providing 3× smaller file sizes compared to PNG.

REGENERATE WEBP IMAGES

You can regenerate all your images safely.

PRODUCT IMAGES	0/20588	0%
CATEGORY IMAGES	0/0	0%
SUPPLIER IMAGES	0/1	0%
STORE IMAGES	0/5	0%
MANUFACTURER IMAGES	0/6	0%

IMPORTANT: la fenêtre ou l'onglet où vous avez démarré le processus de régénération **doit rester ouvert tout au long du processus**. La fermeture interrompt le processus et doit être relancé.

Prise en charge du navigateur WEBP

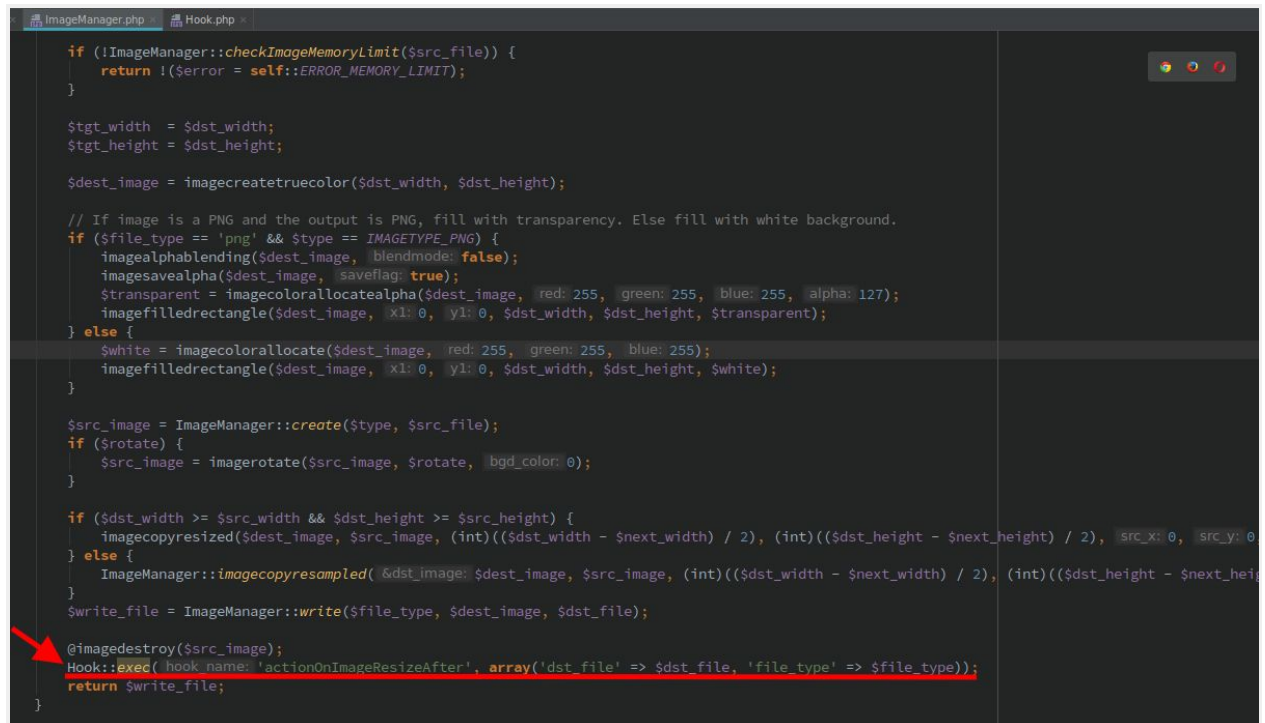
Actuellement, seuls Google Chrome et Opera prennent en charge les images WebP. Bien que d'autres navigateurs, tels que Firefox, Safari ou Internet Explorer, ne prennent actuellement pas en charge le format d'image de manière native, vous n'avez pas à vous inquiéter. Les images de votre site ne seront pas endommagées, même pour les clients utilisant ces navigateurs.

Elles apparaîtront au format PNG ou JPG.

Compatibilité Prestashop 1.6

Comme Prestashop 1.6 n'exécute pas le `actionOnImageResizeAfter` hook, vous devez copier le code suivant dans la `fragment de ImageManager` classe. `redimensionnement` Méthode de, avant l' `$ write_file` renvoyée instruction:

```
Hook :: exec ('actionOnImageResizeAfter', array ('fichier_stst' => $ fichier_stst, 'type_fichier' => $ type_fichier));
```



```

if (!ImageManager::checkImageMemoryLimit($src_file)) {
    return !($error = self::ERROR_MEMORY_LIMIT);
}

$dst_width = $dst_width;
$dst_height = $dst_height;

$dst_image = imagecreatetruecolor($dst_width, $dst_height);

// If image is a PNG and the output is PNG, fill with transparency. Else fill with white background.
if ($file_type == 'png' && $type == IMAGETYPE_PNG) {
    imagealphablending($dst_image, blendmode: false);
    imagesavealpha($dst_image, saveflag: true);
    $transparent = imagecolorallocatealpha($dst_image, red: 255, green: 255, blue: 255, alpha: 127);
    imagefilledrectangle($dst_image, x1: 0, y1: 0, $dst_width, $dst_height, $transparent);
} else {
    $white = imagecolorallocate($dst_image, red: 255, green: 255, blue: 255);
    imagefilledrectangle($dst_image, x1: 0, y1: 0, $dst_width, $dst_height, $white);
}

$src_image = ImageManager::create($type, $src_file);
if ($rotate) {
    $src_image = imagerotate($src_image, $rotate, bgd_color: 0);
}

if ($dst_width >= $src_width && $dst_height >= $src_height) {
    imagecopyresized($dst_image, $src_image, (int)((($dst_width - $next_width) / 2), (int)((($dst_height - $next_height) / 2), $src_x: 0, $src_y: 0);
} else {
    ImageManager::imagecopyresampled( &$dst_image: $dst_image, $src_image, (int)((($dst_width - $next_width) / 2), (int)((($dst_height - $next_height) / 2), $src_x: 0, $src_y: 0);
}
$write_file = ImageManager::write($file_type, $dst_image, $dst_file);

@imagedestroy($src_image);
Hook::exec('hook_name: 'actionOnImageResizeAfter', array('dst_file' => $dst_file, 'file_type' => $file_type));
return $write_file;
    
```