

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 du produit dans Addons PrestaShop)

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

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.

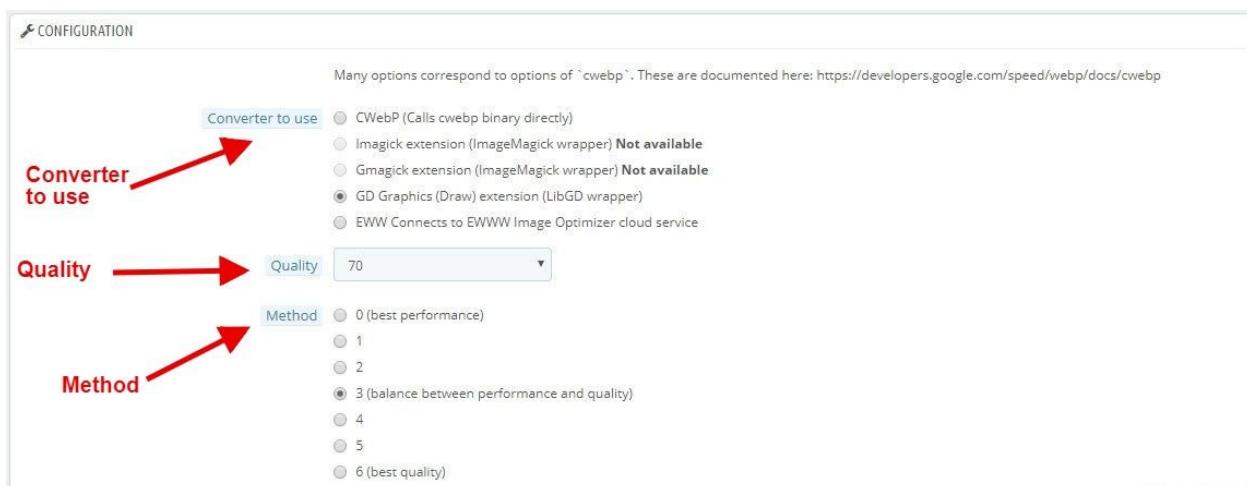
Info: - le module peut être installé à partir du back office du site web, comme tous les autres modules.

Configuration du module

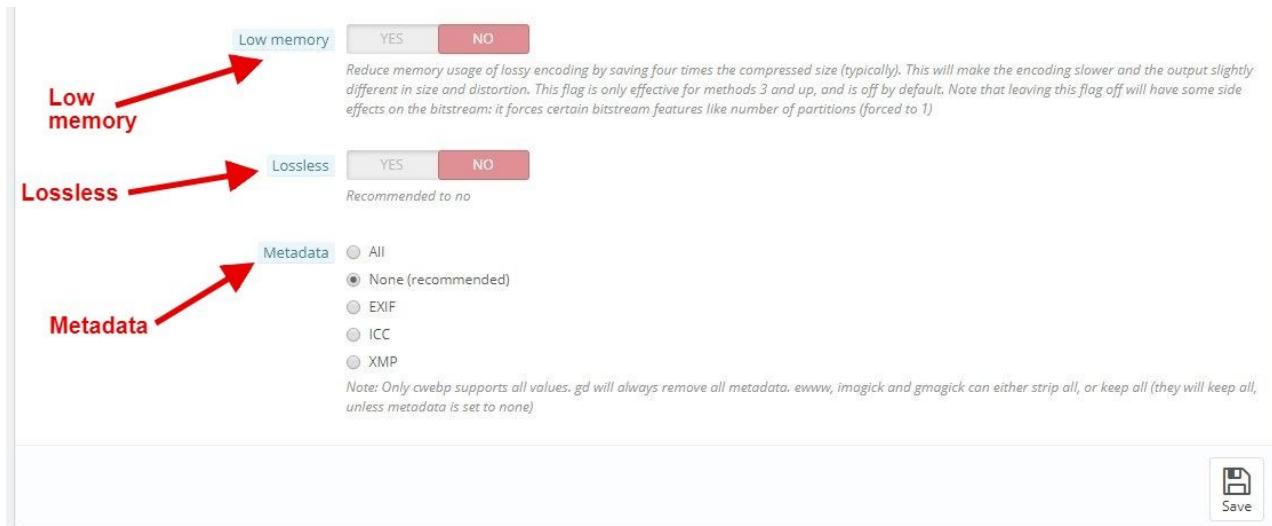
- Une fois l'installation terminée, vous pouvez accéder au module en cliquant sur le **Configurer** bouton.



- **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.



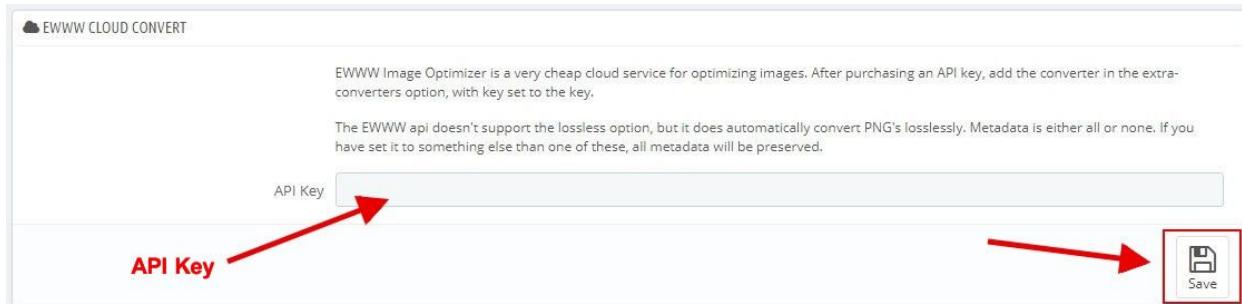
- **Mode démo -** Activez le «mode démo» pendant le processus de génération d'image. Ceci est recommandé pour éviter les erreurs 404 d'images non encore générées. Une fois le processus terminé, vous pouvez désactiver cette option.



EWWW Cloud Convert

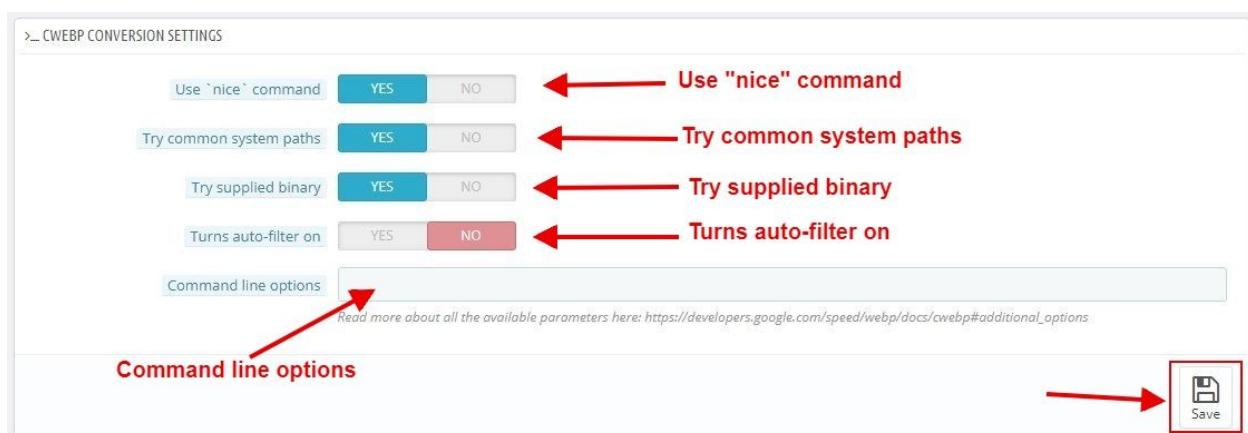
L'optimiseur d'images EWWW est un service de cloud computing très économique pour l'optimisation des images. Après l'achat d'une clé API, ajoutez le convertisseur dans l'option extra-convertisseurs, avec le champ de clé contenant la clé API.

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.



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

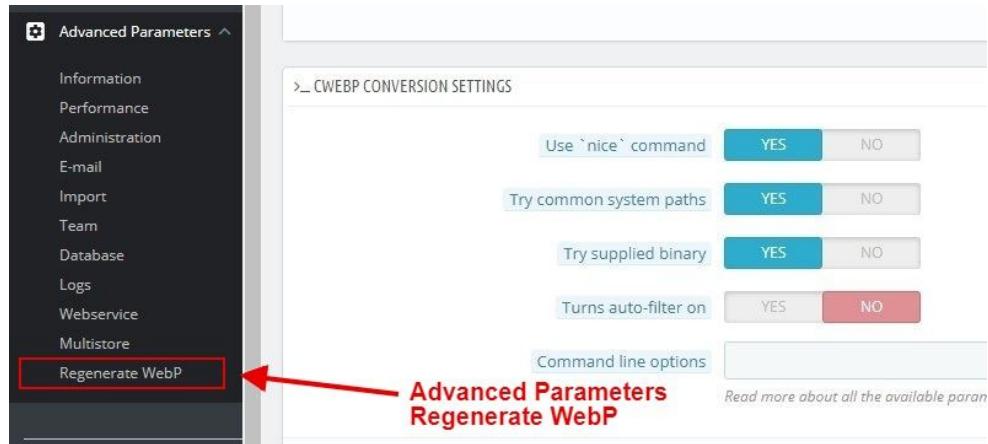


En savoir plus sur tous les paramètres disponibles ici:

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.

The screenshot shows a module titled "REGENARATE WEBP IMAGES". It contains a message: "You can regenerate all your images safely." followed by a list of categories with their current counts:

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

A red arrow points from the text "La fermeture interrompt le processus et doit être relancé." to the progress bar for the "PRODUCT IMAGES" category.

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

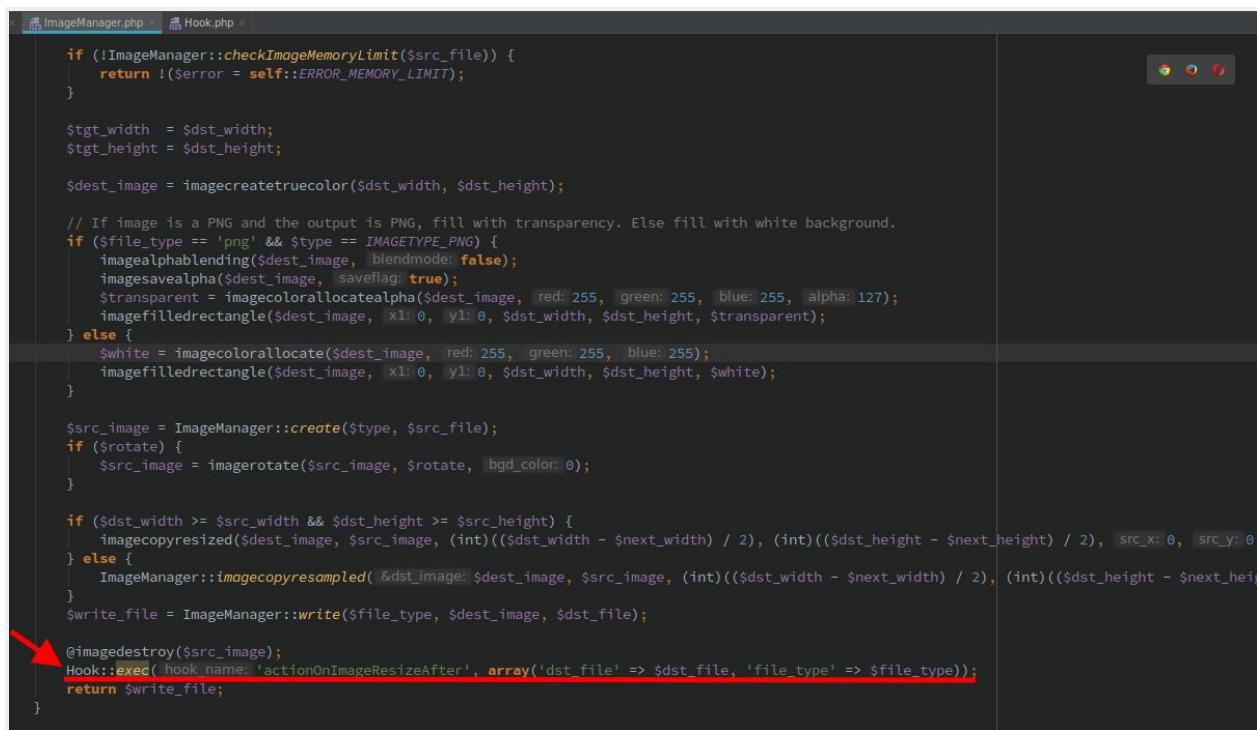
Actuellement, seuls Google Chrome et Opera prennent en charge les images WebP. Bien que d'autres navigateurs, tels que Firefox, Safari ou Internet Explorer, ne prennent actuellement pas en charge le format d'image de manière native, vous n'avez pas à vous inquiéter. Les images de votre site ne seront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

$tgt_width = $dst_width;
$tgt_height = $dst_height;

$dest_image = imagecreatetruecolor($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' && $type == IMAGETYPE_PNG) {
    imagealphablending($dest_image, blendmode: false);
    imagesavealpha($dest_image, saveflag: true);
    $transparent = imagecolorallocatealpha($dest_image, red: 255, green: 255, blue: 255, alpha: 127);
    imagefilledrectangle($dest_image, x1: 0, y1: 0, $dst_width, $dst_height, $transparent);
} else {
    $white = imagecolorallocate($dest_image, red: 255, green: 255, blue: 255);
    imagefilledrectangle($dest_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 >= $src_height) {
    imagecopyresized($dest_image, $src_image, (int)((($dst_width - $next_width) / 2), (int)((($dst_height - $next_height) / 2), src_x: 0, src_y: 0));
} else {
    ImageManager::imagecopyresampled( &$dest_image: $dest_image, $src_image, (int)((($dst_width - $next_width) / 2), (int)((($dst_height - $next_height) / 2), dst_x: 0, dst_y: 0));
}
$write_file = ImageManager::write($file_type, $dest_image, $dst_file);

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

```