

# Traitement des images numériques avec `convert`

Éric Guichard

Novembre 2014

## 1 Introduction

Ces pages présentent de façon succincte un logiciel libre et utilisable sur toute plate-forme : `convert`. Cette documentation en français vaut donc pour tous les systèmes d'exploitation, même si les exemples ont été testés sur Mac et Linux. Un autre logiciel du même type, `exiftool`, sera prochainement décrit. Il est parfois évoqué dans ce texte.

### 1.1 Lecture de ce document

Ces pages ne supposent aucune compétence informatique : il vous suffit de (re)copier les lignes de commande ici proposées pour obtenir le résultat prévu.

Il est possible que dans les commandes à copier à partir du pdf, des apostrophes incurvées ou des retours chariots se soient incrustés (ce souci a été évité, mais sait-on jamais : voir la note de la page 4). En cas de souci, remplacez ces apostrophes par des apostrophes droites et éliminez les retour-chariot dans la fenêtre du *terminal* (celle qui recueille les lignes de commande).

Enfin, ne vous inquiétez pas si certains termes des commandes à copier sont en gras : ce problème est sans incidence sur le résultat, il sera corrigé prochainement.

Vous trouverez à l'URL <http://barthes.enssib.fr/cours/image> ces notes de cours ainsi que les images utilisées comme exemples.

Enfin, certaines précisions techniques sont signalées par un **T!** en marge gauche du texte. Vous n'êtes pas obligé/e de les lire si vous les trouvez trop complexes.

### 1.2 `convert`

`Convert` permet de convertir, réduire, modifier, traiter une photo, ou une série de photos, de scans, voire de pdfs : à peu près tous les formats imaginables sont utilisables. C'est l'outil de travail en ligne du logiciel Imagemagick (son cheval de trait : *the main workhorse*, disent les

Américains). Cf. <http://www.imagemagick.org/script/convert.php>. On pourrait le comparer à des outils comme Gimp, Photoshop, Lightroom. Son intérêt est de fonctionner avec des lignes de commande : on peut lancer un traitement massif en un seul script.

### 1.3 `exiftool`

Exiftool, qui fera prochainement l'objet d'une documentation autonome, permet d'indexer une image, nativement numérique ou issue d'un scan, quel qu'en soit le format. Plus précisément, `exiftool` met des étiquettes à l'intérieur d'une photo : on peut l'utiliser pour supprimer des « méta données » (par exemple la marque du scanner à l'origine d'une copie, la date de création ou le propriétaire d'une image, voire d'un fichier Word), ou en insérer d'autres (légende, orientation de l'image, etc.). Cette méthode iconographique est en vogue dans les agences de presse, mais toujours un peu délicate à utiliser car on ne sait jamais quelles « méta données » un logiciel va ajouter, renommer ou supprimer. La notion d'*instabilité scribale* vaut donc aussi pour les outils de traitement de l'image (comme GraphicConverter, ViewNX, Photoshop, etc.). Il est donc conseillé de toujours garder en mémoire (dans un fichier externe) les informations que l'on insère dans une image afin de pouvoir les réinjecter en cas de besoin.

### 1.4 Découvrir la programmation

Les outils `convert` et `exiftool` fonctionnent très bien avec un langage comme Perl (`exiftool` est écrit en Perl). Aussi l'usage de ces deux logiciels invite-t-il à découvrir la programmation. Par exemple pour uniformiser la taille de certaines photos, les basculer en noir et blanc (NB), et en faire un album web.

En conclusion, un traitement automatisé des images conduit rapidement à utiliser `convert`, `exiftool`, Perl, un *terminal* (pour passer des commandes Unix/Linux, et celles de `convert` et `exiftool`), voire des outils complémentaires pour convertir des formats de fichier que ne reconnaît pas encore `convert` (ex. : `dcraw` pour le DNG).

Sur Linux, vous pourrez visualiser vos images avec `gwenview`. Sur Mac, avec Aperçu (Preview) ou avec de nombreux autres logiciels.

### 1.5 Installations et recommandations

Sur Linux, `convert` et Imagemagick s'installent facilement. Sur Mac, Imagemagick est plus délicat à installer. Je conseille l'usage de Macports (<https://www.macports.org/>, qui requiert lui-même un Xcode à jour (2,5 Go, nécessité d'avoir un compte Apple). Alors un `sudo port install ImageMagick` permet une installation correcte, qui inclut `convert` lui-même. Sinon, `convert` seul s'installe aisément sur Mac.

Cette documentation est avant tout un marchepied pour vous faciliter la découverte de ces outils. Elle n'est pas exhaustive, elle propose des solutions qui peuvent assurément être améliorées. Enfin, les scripts ici proposés sont eux-aussi perfectibles. Sauf erreur, ils sont néanmoins sans bug.

## 2 Convert

### 2.1 Chemin d'accès

**T!**

Suivant les machines, `convert` est installé dans les dossiers `/usr/local/bin/` ou `/opt/local/bin/`. Si la saisie de la commande `convert` génère une réponse du genre `convert: Command not found`, vous pouvez essayer de saisir la commande précédée de son chemin d'accès : `/usr/local/bin/convert` ou `/opt/local/bin/convert`.

### 2.2 Des opérations simples

La syntaxe générale de `convert` peut être décrite ainsi :

```
convert image-initiale -options image-finale
```

En fait, c'est un peu plus compliqué, cf. <http://www.imagemagick.org/Usage/basics/#cmdline>.

#### Convertir une photo couleur en NB

```
convert photopedago.JPG -modulate 100,0 NB.JPG
```

Explication : on met la *saturation* à 0.

#### Adaptation à une taille donnée

```
convert photopedago.JPG -resize 1000000@ photo1Mo.JPG
```

Explication : la photo finale (*photo1Mo.JPG*) sera composée d'au plus un million de pixels (donc assurément moins d'octets en JPG : histoires de compressions). Les rapports largeur/hauteur seront respectés. Cette fonction est très utile pour envoyer des photos par mail, pour les poster sur un site, ou pour réaliser un album en ligne.

#### Changement de format

```
convert photopedago.JPG photo.png
```

Attention, le changement de format peut s'avérer gourmand : ici la photo finale est deux fois plus lourde que l'initiale.

Si on convertit un fichier pdf en JPG, autant d'images que de pages sont produites. La qualité n'est pas toujours au rendez-vous.

**T!**

Pour convertir des formats *raw* (« bruts », ex. DNG), il faut parfois passer par une traduction intermédiaire (ex. : DNG vers PPM, avec *dcraw*).

### Alléger la photo en baissant sa qualité

```
convert photopedago.JPG -quality 50 photobasse-def.JPG
```

Info : la qualité varie de 0 à 100.

### Flouter la photo

```
convert photopedago.JPG -blur 10x3 flou.JPG
```

## 2.3 Combinaison d'options

### Combinaisons simples

```
convert photopedago.JPG -resize 500000@ photopetite.gif
```

La photo finale est au format GIF, elle est de taille réduite.

### Insérer un cercle rouge

```
convert photopedago.JPG -fill red -draw 'circle 100,100 150,150' \
photocercle.JPG
```

Le cercle va du point de coordonnées 100,100 au point de coordonnées 150,150.

**Note** : le « \ » en fin de ligne garantit que les deux lignes seront liées (inutile d'enlever le retour-chariot dans la fenêtre du *terminal*).

### Insérer du texte

```
convert photopedago.JPG -font Bookman-LightItalic -pointsize 48 \
-fill white -draw 'text 1200,1100 "Photo_personnelle" ' \
-draw 'text 800,800 "Photo_personnelle" ' \
photo-texte.JPG
```

Il n'est pas toujours facile de connaître le nom d'une police. Sur un Mac, *a priori* les noms suivants fonctionnent : Helvetica, Monaco, AGaramondPro-Bold, Palatino.

### On peut récidiver simplement

```
convert photopedago.JPG -font Bookman-LightItalic -pointsize 48 \  
-fill white -draw 'text 1200,1100 "Photo_personnelle" ' -draw \  
'text 800,800 "Photo_personnelle" ' photo-texte.JPG
```

### Dessiner un cercle

```
convert -size 30x30 xc:black -fill white -draw 'circle 15,15 5,15' \  
-append cerclesimple.png
```

### Ajout de couleurs/filtres

```
convert photopedago.JPG -colorize 100,200,0 biz.JPG
```

À manier avec précaution (ajoute ou enlève du rouge, vert, bleu).

## 2.4 Travail sur des séries de photos

Notes :

- s'il faut écrire une série de commandes `convert` sur plusieurs lignes, un `\` en fin de ligne facilite la lecture (voir la précédente note page 4) ;
- ne vous inquiétez pas des mots apparaissant en gras dans les commandes qui suivent (comme le *set* ou le *for*).

Exemple :

### Modifier des images en tenant compte de leurs noms

```
convert *.JPG -resize 50000@ -modulate 100,0 \  
-set filename:fname '%t_petiteNB' +adjoin '%[filename:fname].gif'
```

Il n'est pas nécessaire de tout comprendre de cette « formule » pour en faire bon usage ou la détourner.

**Variante : mettre le résultat dans un dossier pré-créé** On suppose qu'un dossier `petitesimages` a été créé.

```
convert *.JPG -resize 50000@ -modulate 100,0 -set filename:fname \  
'%t_petiteNB' +adjoin petitesimages/'%[filename:fname].gif'
```

## 2.5 Vers la programmation

**T!** Attention, cette section est assez technique.

### 2.5.1 Avec un shell

#### Exemple élémentaire

```
mkdir petitesimages
for f in *.JPG
do convert $f -resize 50000@ petitesimages/$f.gif
done
```

### 2.5.2 Avec Perl

Voir le programme `newpgmweb.pl`, qui construit un site/album web homogène à partir de photos d'origines diverses.

Voir aussi le programme `pgm-chgt-nom.pl`, qui modifie les parties de nom des images d'un dossier donné.

Ces deux programmes seront prochainement en ligne.