

MINI-PROJET  
*Gestion de photos à partir de tags*  
Novembre 2012

---

Vous allez réaliser un mini-projet en shell de Bourne. Ce projet est à rendre **au plus tard le dimanche 6 janvier 2013 à 23h59**. Vous devrez fournir les scripts et un mini-rapport (maximum 5 pages) décrivant ces scripts ainsi que des exemples d'utilisation. Vous rendrez ces documents sous forme d'une archive (tar) contenant les fichiers sources des scripts et le rapport au format PDF. Les détails des modalités de cette remise ainsi que de la validation de votre travail vous seront donnés sur la page consacrée à l'UE « systèmes 1 » du wiki enseignement (<http://www.irit.fr/enseignement>).

---

## Sujet

Les fichiers images qui contiennent des photos incluent souvent des informations concernant les photos sous la forme de balises de « métadonnées » qui respectent les spécifications Exif (*Exchangeable image file format*), IPTC (*International Press Telecommunications Council*) ou XMP (*Extensible Metadata Platform*). Dans la suite, nous appellerons ces balises des *tags*. On trouve notamment la date de la prise de vue, le type d'appareil photo, la taille de l'image...

Nous vous proposons d'utiliser ces tags pour gérer les photos. Vous allez donc écrire plusieurs scripts de gestion de fichiers images, certains simples, qui seront utilisés pour en réaliser de plus complexes.

**Précision :** chaque script devra vérifier la validité des paramètres qui lui sont passés, l'accessibilité des fichiers et répertoires qu'il manipule et le fait que les fichiers images contiennent des images (ou du moins les tags associés aux images) ; à chaque appel d'une commande dans un script, il faudra gérer les erreurs éventuelles et afficher des messages clairs pour l'utilisateur.

## Le programme `exiv2`

Le programme `exiv2` (<http://www.exiv2.org>) est une commande que l'on ajoute au système. Elle permet d'accéder aux tags dans un fichier image, pour les afficher, les modifier ou même en créer de nouveaux. La commande `exiv2` n'a été installée que sur le serveur `nbrehat` (accessible depuis l'extérieur grâce à l'adresse `nbrehat.cict.fr`). C'est donc sur ce serveur que vous devez tester vos scripts.

**Remarque :** sur le serveur `nbrehat`, si vous souhaitez afficher des images, vous pouvez utiliser la commande provenant de l'outil ImageMagick (<http://www.imagemagick.org>) :

```
display fichier_photo [fichier_photo ...]
```

ou bien la commande *Eye of GNOME* :

```
eog fichier_photo [fichier_photo ...]
```

Vous allez dans un premier temps consulter la documentation de la commande `exiv2`, accessible en tapant la commande `man exiv2`, et faire des essais en l'utilisant en ligne de commande. Ensuite vous l'utiliserez dans les scripts que vous allez écrire.

## Scripts simples

1. `Date_prise_de_vue.sh` `[-o] fichier_photo [fichier_photo ...]`

Ce script prend en entrée une liste de noms de fichiers et affiche pour chaque fichier, son nom suivi de la date de prise de vue de la photo qu'il contient (tag : `Exif.Photo.DateTimeOriginal`).

L'option `-o` permettra d'afficher la liste des photos dans l'ordre chronologique de prise de vue.

2. `Met_date_dans_nom.sh [-a] fichier_photo`

Ce script prend en entrée un nom de fichier et va ajouter à la fin du nom, avant l'extension, la date de prise de vue de la photo, sous la forme `.AAAA-MM-JJ_hh.mm.ss`

L'option `-a` ne change pas le nom du fichier mais affiche le nom du fichier suivi de `->` suivi du nom du fichier incluant la date.

Par exemple, si la date de la prise de vue de la photo `ma_trombine.jpg` est `2008:05:08 10:15:07`, alors la commande : `Met_date_dans_nom.sh ma_trombine.jpg`

va changer le nom du fichier en : `ma_trombine.2008-05-08_10.15.07.jpg`

et la commande : `Met_date_dans_nom.sh -a ma_trombine.jpg`

affiche : `ma_trombine.jpg -> ma_trombine.2008-05-08_10.15.07.jpg`

sans modifier le nom du fichier.

Si le nom de fichier contient déjà une date, le script affiche un message d'erreur et se termine sans modifier le nom du fichier.

3. `Change_description.sh fichier_photo description`

Ce script prend en entrée un nom de fichier et une chaîne de caractères description. Il va modifier le tag `Exif.Image.ImageDescription` en remplaçant son contenu par la chaîne description.

4. `Change_date_modif.sh fichier_photo`

Ce script prend en entrée un nom de fichier et va modifier sa date de dernière modification (celle affichée avec la commande `ls -l`) selon la date de prise de vue de la photo.

## Scripts complexes

1. `Range_selon_date_et_description.sh [-c | -a] fichier_photo [répertoire]`

Ce script prend en entrée un nom de fichier et en option la désignation d'un répertoire. En utilisant la date de prise de vue de la photo et sa description, il va ranger le fichier dans des sous-répertoires de la forme Année/Mois/Description. Si ces sous-répertoires n'existent pas, il faut les créer. Si le paramètre répertoire est absent, les sous-répertoires appartiennent au répertoire courant, sinon ils appartiennent à répertoire.

L'option `-c` ne déplace pas le fichier mais crée les sous-répertoires.

L'option `-a` ne déplace pas le fichier et ne crée pas les sous-répertoires, mais affiche la désignation complète des sous-répertoires à créer.

Par exemple, si la date de la prise de vue de la photo `ma_trombine.jpg` est `2008:05:08 10:15:07` et sa description `Toulouse`, alors la commande :

`Range_selon_date_et_description.sh ma_trombine.jpg MES_PHOTOS`

va créer (si besoin) dans le répertoire `MES_PHOTOS/` les sous-répertoires `2008/05/Toulouse/` et y déplacer le fichier `ma_trombine.jpg`

et la commande :

`Range_selon_date_et_description.sh -a ma_trombine.jpg MES_PHOTOS`

affiche `MES_PHOTOS/2008/05/Toulouse/` sans créer de sous-répertoire ni déplacer le fichier.

Attention, si la description contient des caractères non alpha-numériques comme l'espace, l'apostrophe, / ou autres, cela peut générer des noms de répertoires difficiles, voire impossibles à gérer. Le script prendra donc soin de remplacer ces caractères par d'autres comme par exemple le caractère souligné `_`

Si la description est la chaîne vide, alors on ne crée pas le sous-répertoire correspondant et le fichier est déplacé dans le sous-répertoire Année/Mois/

2. `Range_Mes_Photos.sh [-d répertoire] fichier_photo [fichier_photo ...]`

Ce script prend en entrée une liste de noms de fichiers et en option la désignation d'un répertoire. Il va ranger ces fichiers selon leur date de prise de vue et leur description à l'aide du script précédent.