



## Installer un serveur de numérisation, SANE

Le 7 août 2008 à 12:35.

L'objectif ici est d'installer le scanner USB en utilisant les outils du projet SANE. Une procédure très simple en soit avec juste quelques "pièges" qu'il est utile d'éviter pour ne pas perdre de temps.

### Le logiciel SANE

Un petit mot sur SANE tout d'abord. Ce projet apporte à un système unix un support unifié de toutes les périphériques d'acquisition d'image. Ce va de la WebCam à la carte TNT en passant bien évidemment par les scanners. Outre ce support, SANE permet de publier un scanner sur votre réseau et ainsi de le rendre utilisable sur d'autres machines. A ce titre SANE est l'équivalent de CUPS mais pour l'acquisition. Le support des WebCam et autre carte TV est en soit anecdotique, il s'agit d'un driver générique qui utilise le système V4L (Vidéo4Linux). Le support des scanners est le vrai métier de SANE.

A noter enfin que le partage des scanners publiés par SANE ne se limite pas au monde Unix. Vous pouvez les utiliser avec [Max/OSX](#) [1], mais aussi sous [Windows](#) [2].

Ce qui suit n'est pas spécifique à un scanner, je prends juste le mien comme exemple pour illustrer la démarche.

### Les drivers Epson

Pour commence un petit point sur les drivers. Epson fait parti de ces fabricant à disposer de drivers pour Linux. C'est assez rare pour être salué et c'est d'ailleurs ce qui fait que je n'achète dans ce domaine (numérisation, impression) que du matériel Epson. la RX640 n'échappe pas à la règle et son drivers se trouve sur le site de la société [Avasys](#) [3] qui semble être une filiale dédiée au développement Linux. Il sont fournis AVEC les sources, sous licence GPL, ce qui est encore plus rare (A titre d'information, les pilotes Linux des cartes nVidia ne sont fournis qu'en binaire fermés). Un aspect qui n'est pas négligeable car du coup le pilote Epson pour la RX640 est déjà intégré dans Saned. Il n'y a donc rien à faire de ce côté qu'installer SANE.

### Installation de SANE

Comme toujours, nous commençons par une installation de paquets (urpmi ou apt-get sane). Ceci fait, la première chose à vérifier est la présence sur le serveur du scanner (qui doit être branché 😊). Pour cela, lancer la commande :

```
root# sane-find-scanner
(... )
found USB scanner (vendor=0x04b8, product=0x081c) at libusb:003:004
(... )
root#
```

Parfait, le scanner est visible. Si ce n'était pas le cas, tentez un `lsusb` pour vérifier s'il est connu du module USB de Linux. Maintenant, nous allons déterminer si SANE a bien un driver pour notre scanner. Pour cela, lancez la commande :

```
root# scanimage -L
device `epkowa:libusb:003:004' is a Epson Stylus Photo Rx640/RX650 flatbed scanner
root#
```

(Re)parfait. Comme je le disais plus haut, le driver Epson GPL (alias epkowa) est déjà intégré dans Sane. A ce stade le scanner est reconnu en local. Nous allons maintenant publier cette périphérique sur le réseau.

### Publication sur le réseau

Première chose à faire, vérifier que le port de SANE, 6566, est présente dans la liste des ports connus par votre serveur. Pour cela, éditer le fichier `/etc/services` et rechercher la chaîne 6566. Si cette ligne existe, vérifiez qu'elle correspond à ce qui suit, sinon, ajoutez là :

```
| sane 6566/tcp saned # SANE network scanner daemon
```

Le démon `saned` n'est pas lancé en permanence sur le serveur. Ce serait idiot de prendre de la mémoire pour un service qui ne sert pas tout le temps. A la place, est utilisé le démon `xinetd` [4]. Ce dernier va seulement se placer en écoute du port de SANE et le lancer lorsque quelqu'un va chercher à s'y connecter.

Il faut donc éditer le fichier qui a du être créé à l'installation de SANE, `/etc/xinetd.d/sane` et vérifier qu'il correspond à cela :

```
| # default: off
```

```
# description: The sane server accepts requests \
# for network access to a local scanner via the \
# network.
service sane
{
    disable = no
    port = 6566
    socket_type = stream
    wait = no
    groups [5] = yes [6]
    user = saned
    group = saned
    server = /usr/sbin/saned
}
```

Le point à valider est que la première ligne soit bien `service sane` et non pas `service sane-port` comme ce fût le cas pour moi. Sans cela, `xinetd` n'arrive pas à retrouver le nom du service et désactive `sane`.

Une fois ceci validé, relancez `xinetd` via un `service xinetd restart` et vérifiez que le port est bien en écoute par la commande suivante : `##netstat -anlp | grep 6566 tcp 0 0 0.0.0.0:6566 0.0.0.0:* LISTEN 7580/xinetd`

A ce stade le scanner est publié, vous pouvez donc aller sur une autre machine pour vérifier son bon fonctionnement.

## Paramétrage du poste client

Sur le poste client, SANE doit aussi être installé. Ceci fait, il n'est nécessaire que de modifier le fichier `/etc/sane.d/net.conf` pour y ajouter le nom de la machine à laquelle est connecté votre scanner. Imaginons qu'elle s'appelle `serveur_scanner`, vous devez donc avoir dans `/etc/sane.d/net.conf` la ligne :

```
| serveur_scanner
```

Une fois le fichier sauvé, il suffit de vérifier que le scanner est bien visible en réutilisant la commande `scanimage` `##scanimage -L device `net:serveur_scanner:epkowa:libusb:003:004` is a Epson Stylus Photo Rx640/RX650 flatbed scanner`

Le scanner est donc bien visible et prêt à être utilisé, par exemple via la commande `scanimage` elle-même :

```
| scanimage --mode color --resolution 150 -d net:serveur_scanner:epkowa:libusb:003:004
> mon_image
```

Et voilà.

## Petit script de scan

Pour terminer, voyons comment utiliser `scanimage` pour automatiser des numérisations nombreuses (genre un acte de propriété de 50 pages, au hasard 😊)

```
#!/bin/sh
# paramètres
target_name="$1" # Nom du fichier final
page_count="$2" # optionnel, le nombre de feuilles à numériser

# on supprime le dossier temporaire de numérisation, on le recrée et on s'y place
if [ -d scans ] ; then
    rm [7] -rf scans
fi
mkdir [8] scans
cd scans

# utilisation de scanimage pour déterminer l'adresse de notre scanner
scanner_usb_id=$( scanimage -L | grep [9] epkowa | cut [10] -d "\`" -f2 | cut [10] -d "'" -f1)

# options de numérisation par défaut
scanimage_options="--mode color --resolution 150 -d $scanner_usb_id"

# Choix entre 1 et plusieurs pages.
if [ ! -z $page_count ] ; then
```

```

    echo "Génération des fichiers sur le modèle $TIME_STAMP"
    scanimage $scanimage_options --batch="$target_name %d" --batch-count $page_count -
-batch-prompt
else
    scanimage $scanimage_options > $target_name
fi

# conversion de toutes les images en JPEG
for file [11] in * ; do
    echo "Conversion de $file en jpg"
    convert -quality 80 "$file" "$file.jpg"
    rm [7] -rf "$file"
done

# Si nous avons plus d'une page, on converti les JPEG en un seul document PDF
if [ ! -z $page_count ] ; then
    convert * "../$target_name.pdf"
else
    mv [12] -f * ..
fi
cd ..

# ménage
rm [7] -rf scans

```

Il a l'air de rien ce script mais il est très très pratique. Lancé avec un nom de fichier en paramètre, il va scanner et générer le fichier JPEG correspondant.

Et si vous indiquez en plus du nom de fichier, un nombre de pages à scanner, il fabriquera cette fois un PDF de toutes les pages scannées.

## Conclusion

SANE est un projet mature qui fonctionne merveilleusement bien. Si vous n'avez pas envie de jouer à la ligne de commande, il existe de très nombreux frontaux graphiques qui le prennent en charge. Citons Kooka sous KDE, XSane ou le tout jeune projet [GnomeScan](#) [13].

<http://artisan.karma-lab.net/node/1159>  
(C) artisan numerique - CC BY-SA

### Liens:

- [1] <http://www.ellert.se/twain-sane/>
- [2] <http://sanetwain.ozuzo.net/>
- [3] <http://www.avasys.jp/english>
- [4] <http://artisan.karma-lab.net/node/88>
- [5] <http://pwet.fr/man/linux/commandes/groups>
- [6] <http://pwet.fr/man/linux/commandes/yes>
- [7] <http://pwet.fr/man/linux/commandes/rm>
- [8] <http://pwet.fr/man/linux/commandes/mkdir>
- [9] <http://pwet.fr/man/linux/commandes/grep>
- [10] <http://pwet.fr/man/linux/commandes/cut>
- [11] <http://pwet.fr/man/linux/commandes/file>
- [12] <http://pwet.fr/man/linux/commandes/mv>
- [13] <http://live.gnome.org/GnomeScan>