

# youtube-dl + vlcopen

Lire directement les vidéos youtube (& cie) dans VLC

mardi 7 avril 2015, par [placido](#)

Vous le savez comme moi, flash c'est le mal. Et voici LA solution qui vous fera franchir le pas d'un bannissement (espérons permanent) du vil greffon sans pour autant faire une croix sur les vidéos des plateformes d'hébergement (youtube, dailymotion, vimeo, etc).

Cela consiste en l'emploi coordonné de **deux scripts** qui vont vous permettre de lire directement une vidéo distante dans votre lecteur habituel (VLC, mplayer ou autre).

## 1) Installation de youtube-dl

Sous GNU-Linux, la méthode que je trouve la plus simple est de passer par le gestionnaire d'environnement python pip.

1. **sudo apt-get install** python-pip
2. **sudo pip install** youtube-dl

Veillez à mettre à jour l'application de temps en temps, car les plugins de gestion par plateforme sont nombreux et les modifications fréquentes.

1. **sudo pip install -U** youtube-dl

## 2) Lancement illico grâce à vlcopen.sh

[youtube-dl](#) est un super logiciel, efficace dans bien des situations, mais son utilisation en ligne de commande s'avère à la longue un peu réhhibitoire pour une utilisation en tant que simple utilisateur.

D'où le recours à un second script, mis à disposition par [swarminglogic](#), qui va apporter un peu de souplesse d'utilisation et ouvrir la vidéo encore plus vite que si vous y aviez accédée directement depuis la page youtube, grâce à un raccourci clavier.

1. `#!/bin/bash`
- 2.
3. **if [ \$# -eq 1 ] ; then**
4. `path=$1`
5. **else**
6. `clipboard=$(xclip -selection clipboard -o)`
7. **if [ \$(<&2**
8. **else**
9. `notify-send "Bad clipboard path: $clipboard"`
10. **fi**
11. **exit 1**

```

12. fi
13. fi
14.
15. # $1 processid, $2 tmpdir $3 logfile
16. function cleanup {
17.     echo "-----"
18.     echo "Cleaning up"
19.     kill -9 $1 &> /dev/null
20.     if [ -n "$2" ]; then rm $2/* ; fi
21.     if [ -n "$2" ]; then rmdir $2 ; fi
22.     if [ -n "$3" ]; then rm $3 ; fi
23. }
24.
25.
26. echo "Opening $path"
27. tmpvlcopen=/tmp/vlcopen/
28. [ ! -d $tmpvlcopen ] && mkdir $tmpvlcopen
29. tmpdir=$(mktemp -d --tmpdir=$tmpvlcopen)
30.
31. # If the tmpdir path doesn't include "/tmp/vlcopen" something went bad!
32. if [ "${tmpdir/$tmpvlcopen}" = "$tmpdir" ]; then
33.     echo "Aborting: failed to create proper temporary directories" >&2
34.     exit 1
35. fi
36.
37. cd $tmpdir
38. echo "Temporary directory: $tmpdir"
39. echo "Getting video: $path"
40. downloadlog=$(tempfile)
41. (youtube-dl --restrict-filenames --no-part "$path" &> $downloadlog)&
42. processId=$!
43. echo "ProcessID: $processId"
44. echo "Waiting for file creation"
45.
46. count=0
47. while [ $(ls -1 | wc -l) -eq 0 ]; do
48.     # check if process is done.
49.     if ! kill -0 $processId &>/dev/null ; then
50.         if [ -t 1 ] ; then
51.             echo "-----">&2
52.             echo "Error: youtube-dl finished without output." >&2
53.             echo "Error: Bad URL?" >&2
54.             echo "----- Download Log -----">&2
55.             cat $downloadlog >&2
56.         else
57.             notify-send "Bad URL: $path $(cat $downloadlog)"
58.         fi
59.         cleanup $processId "$tmpdir" "$downloadlog"
60.         exit 0
61.     fi
62.     sleep 0.5

```

```

63. ((count++))
64. if [ $count -eq 20 ] ; then
65.     if [ -t 1 ] ; then
66.         echo "Timed out" >&2
67.     else
68.         notify-send "Timed out"
69.     fi
70.     exit 0
71. fi
72. done
73.
74. sleep 0.1
75. echo "Opening file: $(ls -1 | head -n 1)"
76. vlc "$(ls -1 | head -n 1)" &>/dev/null
77. cleanup $processId "$tmpdir" "$downloadlog"

```

### [Télécharger](#)

Ce script se comporte comme une capsule pour youtube-dl. Il va analyser le contenu du presse-papier, et s'il détecte une url prise en charge, va créer un répertoire temporaire, y télécharger la vidéo, l'ouvrir avec VLC après quelques instants et supprimer automatiquement tous les fichiers temporaires à la fermeture.

Il suffit d'enregistrer le script (sous le nom `vlcopen.sh` par exemple) dans un répertoire inclus dans votre `$PATH`, le rendre exécutable (`chmod +x vlcopen.sh`) et d'associer un raccourci clavier à son exécution.

Typiquement sous openbox, j'ai rajouté une section comme suit à mon `rc.xml`.

```

2.
3.
4.   true
5.   vlcopen
6.
7.   vlcopen.sh
8.

```

Dès lors, dans les faits, lorsque je veux regarder une vidéo, je fais un *clic-droit* > *copier l'url de la page* soit sur la page courante, soit sur le lien à analyser, puis `super + y` (oui j'ai choisi `y` pour "youtube" mais à vous d'adapter au besoin), et voilà !

Seul bémol, si la vidéo est longue et que vous êtes impatient, vous ne pourrez pas avancer directement la lecture à un point qui n'a pas encore été téléchargé.

Voir en ligne : [github.com/swarminglogic](https://github.com/swarminglogic)