



Python server with web interface: <https://gitlab.com/RemixDev/deemix-webui>  
<https://gitlab.com/RemixDev/deemix-gui/-/tree/main>

Standalone GUI, Server or one-file Server for Linux/Win/Mac x86: <https://download.deemix.app/0/>

Note: all of the above require an arl = arl cookie after logging into (free) deezer account. i.e. F12 in Firefox, storage, cookies, arl, then copy the content. The cookie is valid for 3 months. Entering the arl cookie in the web interface stores it per client. Storing it in ~/.config/deemix/.arl and using -serverwide-arl flag can use the same for all clients.

Linux install/update of server with static libraries:

### [update-deemix.sh](#)

```
#!/bin/bash
echo "Downloading latest version..."
wget https://download.deemix.app/server/linux-x64-latest
if [ -f "linux-x64-latest" ];
then
    echo "Backing up old version..."
    sudo rm -rf /opt/deemix-server-old
    sudo mv /opt/deemix-server /opt/deemix-server-old
    sudo mkdir /opt/deemix-server
    sudo mv linux-x64-latest /opt/deemix-server/deemix-server
    sudo chmod 755 /opt/deemix-server/deemix-server
    echo "Restarting deemix server..."
    systemctl --user restart deemix
    echo "Done!"
else
    echo "Download failed, not updating!"
fi
```

Linux user systemd service for server if extracted to /opt/deemix-server/:

```
mkdir -p .config/systemd/user/
vi .config/systemd/user/deemix.service
```

```
[Unit]
Description=Deemix Server

[Service]
ExecStart=/opt/deemix-server/deemix-server --host 0.0.0.0 --serverwide-arl
Restart=always
RestartSec=10

[Install]
WantedBy=default.target
```

```
systemctl --user enable deemix
```



## daemon

Automatically monitor and/or download new releases of artists.

<https://github.com/digitalec/daemon>

## yt-dl / youtube-dl

Youtube Download (lots of supported sites: <https://yt-dl.org/supportedsites.html> ) <https://yt-dl.org/>  
<https://github.com/ytdl-org/youtube-dl>

Works with youtube, soundcloud and many more

Install/upgrade not via apt or wget/curl but using:

```
sudo -H pip install --upgrade youtube-dl
```

Fixes to individual extractors not in official release yet can be applied as in the following example:

```
wget
https://raw.githubusercontent.com/ytdl-org/youtube-dl/d42b7b6c5e927a8e4e82be96a551b5054e445974/youtube_dl/extractor/redtube.py
sudo mv /usr/local/lib/python3.8/dist-packages/youtube_dl/extractor/redtube.py /usr/local/lib/python3.8/dist-packages/youtube_dl/extractor/redtube.py_backup
sudo mv redtube.py /usr/local/lib/python3.8/dist-packages/youtube_dl/extractor/redtube.py
```

Workaround/Tip: per-site configuration files <https://github.com/ytdl-org/youtube-dl/issues/29523>

Version 2021.12.17 requires the following youtube patch as of February 2023:

<https://github.com/ytdl-org/youtube-dl/issues/31530#issuecomment-1433470254> Change youtube.py (Debian package /usr/lib/python3/dist-packages/youtube\_dl/extractor/youtube.py) (PIP install /usr/local/lib/python3.8/dist-packages/youtube\_dl/extractor/youtube.py) change:

```
'uploader_id': self._search_regex(r'/(?:(channel|user)/([^\/?&#]+)',
owner_profile_url, 'uploader id') if owner_profile_url else None,
```

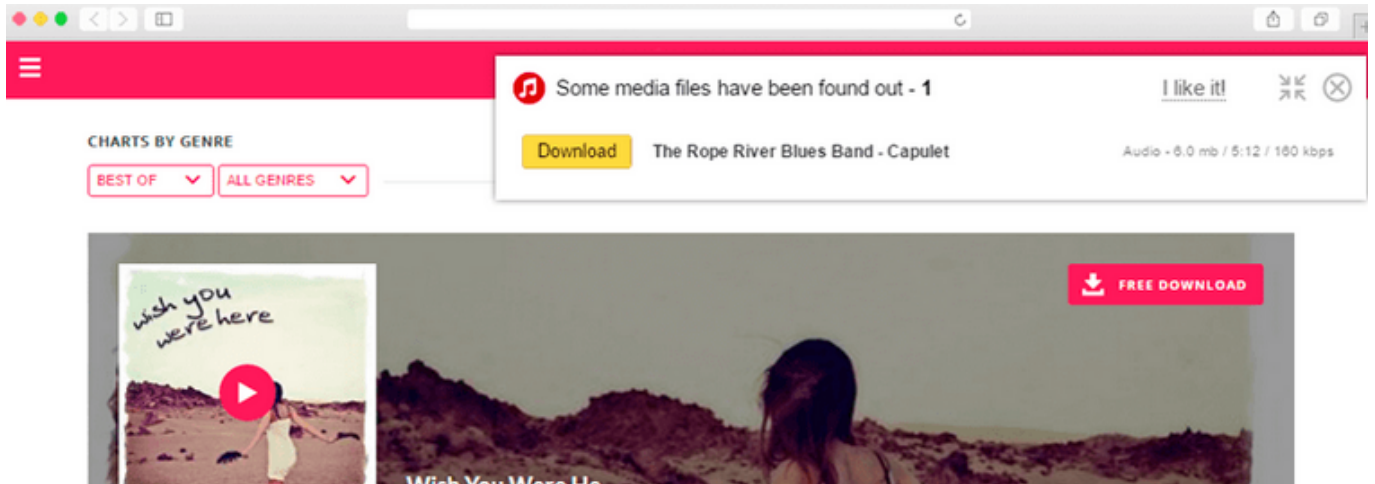
to:

```
'uploader_id': self._search_regex(r'/(?:(channel/|user/|@)([^\/?&#]+)',
owner_profile_url, 'uploader id', default=None),
```

## mediasave Chrome

Mediasave Browser addon (Chrome) <https://mediasave.ru/en/>

<https://chrome.google.com/webstore/detail/mediasave-download-music/hfgbpkkdodfihabamnkhoaea/mkdhnoec>

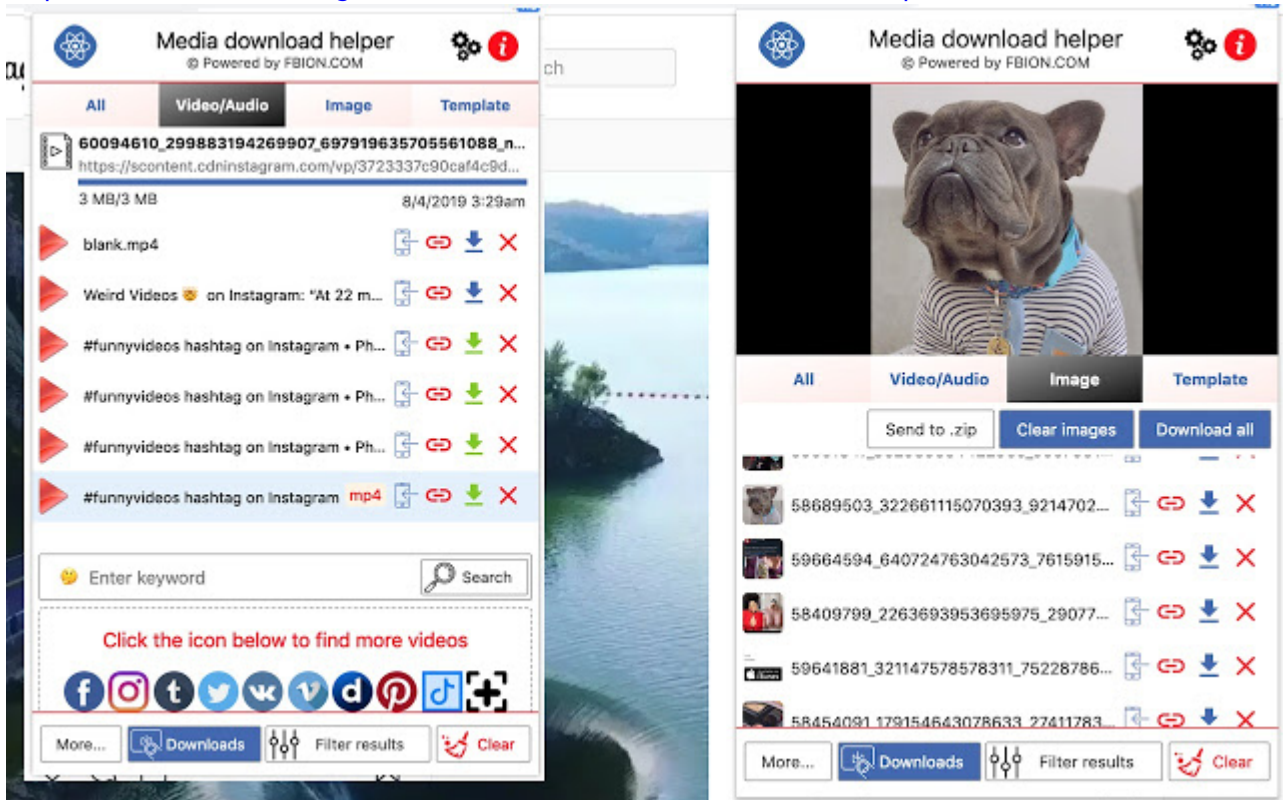


Works with unprotected media

## Media Download Helper Firefox

Media Download Helper (Firefox)

<https://addons.mozilla.org/en-US/firefox/addon/media-download-helper/>



Uses external service to record the stream. No Filenames/ID3tags preserved!

## Instagram downloader

<https://github.com/instaloder/instaloder>

## JDownloader

<https://jdownloader.org/download/index>

## Screen Recording

### SimpleScreenRecorder

```
sudo apt-get install simplescreenrecorder
```

## Video Editing

### OpenShot

Stable PPA (Contains only official releases)

```
sudo add-apt-repository ppa:openshot.developers/ppa
sudo apt update
sudo apt install openshot-qt python3-openshot
```

```
sudo apt-get install openshot
```

### LosslessCut

<https://github.com/mifi/lossless-cut>

[losslesscut.sh](#)

```
#!/bin/bash
if [ "$LOGNAME" != "root" ]
then
    echo "LosslessCut updater needs to be run as root!"
    exit
fi

#get latest LosslessCut version number
latest_web=$(curl https://api.github.com/repos/mifi/lossless-cut/releases/latest -s|sed -n 's/"tag_name":
"v\([0-9]\.[0-9]*\.[0-9]*\)"/,\1/p'| sed -e 's/^[[:space:]]*//')
#convert version numbers to integers for comparison
```

```

web=$(echo "$latest_web" | sed 's/\././g')

if [ -f /opt/losslesscut/version.txt ];
then
    current_ver=$(cat /opt/losslesscut/version.txt)
    #convert version numbers to integers for comparison
    cur=$(echo "$current_ver" | sed 's/\././g')
fi

if [[ "$cur" == "" ]];
then
    cur=0
fi

#download and install only if web version greater than current version
#if main web version doesn't exist, then install new main version
if [[ ${web:0:8} -gt ${cur:0:8} ]];
then
    echo "Current LosslessCut version $current_ver, upgrading to
version $latest_web"
    if [ ! -d /opt/losslesscut ];
    then
        mkdir /opt/losslesscut
    fi
    cd /opt/losslesscut
    if [ -f LosslessCut-linux-x86_64.AppImage ];
    then
        mv LosslessCut-linux-x86_64.AppImage LosslessCut-linux-
x86_64.AppImage.old
    fi
    wget "https://github.com/mifi/lossless-
cut/releases/download/v$latest_web/LosslessCut-linux-x86_64.AppImage"
    chmod 755 /opt/losslesscut/LosslessCut-linux-x86_64.AppImage
    echo "$latest_web" > /opt/losslesscut/version.txt

    if [ ! -f /usr/share/applications/losslesscut.desktop ];
    then
        wget https://github.com/mifi/lossless-
cut/raw/master/src/icon.svg
        cat >> /usr/share/applications/losslesscut.desktop <<EOD
[Desktop Entry]
Name=LossLessCut
Exec=/opt/losslesscut/LosslessCut-linux-x86_64.AppImage
Comment= The swiss army knife of lossless video/audio editing
Terminal=false
Icon=/opt/losslesscut/icon.svg
Type=Application
Categories=Graphics;
Hidden=false
NoDisplay=false
EOD

```

```
update-desktop-database
xdg-desktop-menu forceupdate
xdg-desktop-menu install --novendor
/usr/share/applications/losslesscut.desktop

else
echo "Current Version $current_ver, available version $latest_web"
echo "No upgrade necessary"
fi
```

## Saving DRM streams

<https://forum.videohelp.com/threads/415503-%5Brelease%5D-AllHellGui>

<https://github.com/vinefeeder/HellYes>

```
mkdir hellyes6
cd hellyes6
wget https://files.videohelp.com/u/301890/hellyes6.zip
unzip hellyes6.zip
rm hellyes6.zip
cd HellYesGui
python -m venv venv
venv/bin/pip install -r requirements.txt
```

download the following 4 and copy binaries to any directory in the path, e.g. to ~/.local/bin # uses N\_m3u8DL-RE, ffmpeg, mkvmerge and mp4decrypt # see [https://github.com/nilaoda/N\\_m3u8DL-RE](https://github.com/nilaoda/N_m3u8DL-RE) # see <https://www.videohelp.com/software/ffmpeg> # see <https://www.bento4.com/downloads/> # see <https://www.videohelp.com/software/MKVToolNix>

Start it using the venv python interpreter:

```
venv/bin/python allhell3gui.py
```

Usage as following:

- open debug console in browser (F12) and open network tab
- add regex:widevine|acquire|license|mpd to the search
- open <https://www.channel4.com/> or other desired video page and play the video
- copy the found GET mpd entry and paste into the MPD URL field
- copy the found POST entry as cURL and paste into the cURL of License Request field
- add a video name
- press get Keys
- press download video

From:

<http://wuff.dyndns.org/> - **Wulf's Various Things**

Permanent link:

<http://wuff.dyndns.org/doku.php?id=howto:save-streams&rev=1740399118>

Last update: **2025/02/24 12:11**

