

# **MyScreenRecorder**

**To capture your desktop screen and audio devices.**

## Table des matières

1 Context.....	3
2 Software architecture.....	3
2.1 ffmpeg.....	3
2.2 openjdk11.....	3
2.3 openJFX11.....	4
2.4 myscreenrecorder.....	4
3 Procedure.....	5
3.1 Download Github archive.....	5
3.2 Creation of application folder and extraction.....	5
3.3 Creation of directory for video output.....	6
3.4 Configuration of audio devices.....	6
3.4.1 Microphone case and built-in speakers.....	6
3.4.2 Headset microphone case.....	7
4 Full screen recording.....	9
5 Recording rectangular area.....	10

# Introduction

## 1 Context

Under Windows, I could not (or did) find simple free software to make screenshots and audio as video. Very useful for tutorials for example.

There are some free but the recording time is limited or there is a watermark inserted at the beginning and end of the recording. Others generate uncompressed (very large) avi files with a size limit of 2 GB.

Under Linux, I know at least 2:

- [RecordMyDesktop](http://recordmydesktop.sourceforge.net/downloads.php) => <http://recordmydesktop.sourceforge.net/downloads.php>
- [SimpleScreenRecorder](https://www.maartenbaert.be/simplescreenrecorder/) => <https://www.maartenbaert.be/simplescreenrecorder/>

Although my software can run under Linux, I will focus instead on its use on Windows. I provide all the operating modes and scripts to make it work also under Linux, without details.

## 2 Software architecture

Important: I will describe an installation of **myscreenrecorder** under the directory of your choice :

**C:\opt\myscreenrecorder**

**D:\opt\myscreenrecorder**

If we respect the names described below, we will not have to touch the launch scripts.

### 2.1 ffmpeg

This is the heart of the myscreenrecorder tool, the ffmpeg video processing framework is also capable of capturing screen and audio, but online commands can become very complex.

For the Windows version, it is necessary to take the static binary version (ie including all the libraries) => The download site is [here](https://ffmpeg.zeranoe.com/builds/win64/static/ffmpeg-latest-win64-static.zip) => <https://ffmpeg.zeranoe.com/builds/win64/static/ffmpeg-latest-win64-static.zip>

Note that with Linux, it is better to use the ffmpeg package specific to your distribution. I had problems with the static version.

Unzip under **\opt** (under C: or D: as appropriate); the directory must be called **\opt\ffmpeg-latest-win64-static**

### 2.2 openjdk11

I'm using openjdk 11.0.2 for this version of **myscreenrecorder** to download [here](https://download.java.net/java/GA/jdk11/9/GPL/openjdk-11.0.2_windows-x64_bin.zip) => [https://download.java.net/java/GA/jdk11/9/GPL/openjdk-11.0.2\\_windows-x64\\_bin.zip](https://download.java.net/java/GA/jdk11/9/GPL/openjdk-11.0.2_windows-x64_bin.zip)

Unzip under **\opt** (under C: or D: as appropriate); the directory must be called **\opt\jdk-11.0.2**

## 2.3 openJFX11

I'm using openJFX-SDK 11.0.2 for this version of **myscreenrecorder** to download [here](http://gluonhq.com/download/javafx-11-0-2-sdk-windows/) =>  
<http://gluonhq.com/download/javafx-11-0-2-sdk-windows/>

Unzip under **\opt** (under C: or D: as appropriate); the directory must be called **\opt\javafx-sdk-11.0.2**

## 2.4 myscreenrecorder

This is the current application which is a **ffmpeg** graphic front end for recording the screen and possibly audio through a microphone or through the speakers used (headphones or pc specific).

In addition to the software already mentioned, I also use a library (provided with the github package) interface JNI **JnativeHook** available [here](https://github.com/kwhat/jnativehook/releases/download/2.1.0/jnativehook-2.1.0.zip) =>  
<https://github.com/kwhat/jnativehook/releases/download/2.1.0/jnativehook-2.1.0.zip>

# INSTALLATION

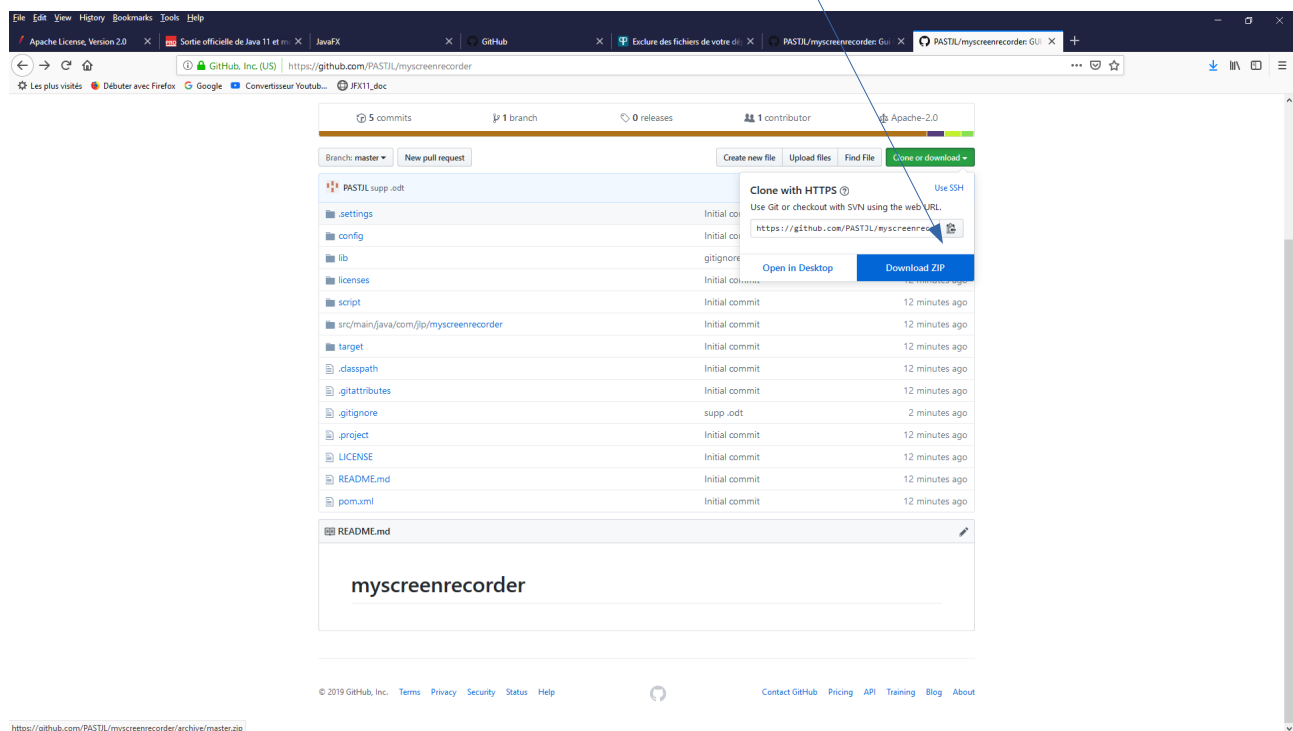
## 3 Procedure

It is assumed that **ffmpeg**, **openjdk** and **openjfx** have been installed according to paragraph 2 above under the directory ( **C:** ou **D:**) \opt\

### 3.1 Download Github archive

Go to the site [Github](https://github.com) to retrieve the zip archive of the project :

<https://github.com/PASTJL/myscreenrecorder>



### 3.2 Creation of application folder and extraction

Create a directory (C: ou D:) \opt\myscreenrecorder

Extract from downloaded zip file ( **myscreenrecorder-master.zip**) the directories :

- **lib**
- **script**
- **config**

- **licences**
- **manuals**

into the directory (C: or D:)\opt\myscreenrecorder

Create a shortcut on the desktop of the file :

(C: or D:)\opt\myscreenrecorder\script\myscreenrecorder.cmd

and associate the icon (C: or D:)\opt\myscreenrecorder\script\myscreenrecorder.ico

### 3.3 Creation of directory for video output

Create a directory (C: or D:)\opt\screenRecordVideos

### 3.4 Configuration of audio devices

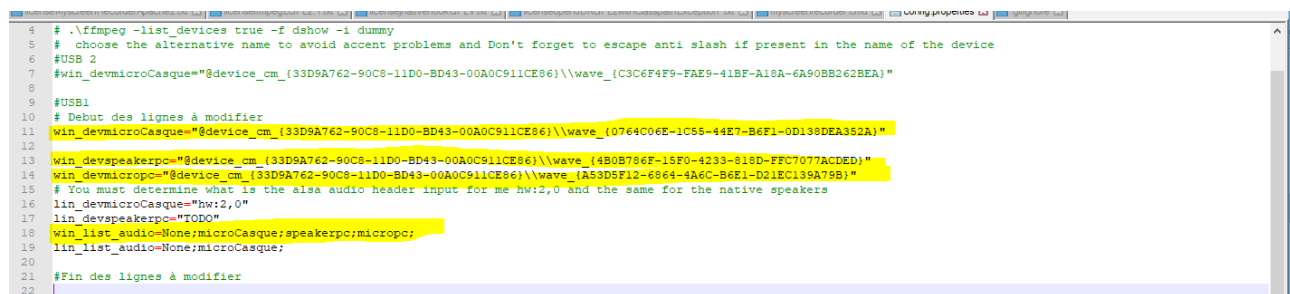
This is the most delicate part of the installation.

In my case in addition to the microphone and speakers built into the pc, I use a headset with microphone.

There are therefore several possible interfaces, but only one must remain active during the recording.

To prevent errors, make a backup copy of the file \opt\myscreenrecorder\config\config.properties

With a text editor (notepad, wordpad ...) open the file \opt\myscreenrecorder\config\config.properties



```

4 # .\ffmpeg -list_devices true -f dshow -i dummy
5 # choose the alternative name to avoid accent problems and Don't forget to escape anti slash if present in the name of the device
6 #USB 2
7 #win_devmicroCasp="@device_cm_{33D9A762-90C8-11D0-BD43-00A0C911CE86}\\wave_{C3C6F4F9-FAE9-41BF-A18A-6A90BB262BEA}"
8
9 #USB1
10 # Debut des lignes à modifier
11 win_devmicroCasp="@device_cm_{33D9A762-90C8-11D0-BD43-00A0C911CE86}\\wave_{0764C06E-1C55-44E7-B6F1-0D138DEA352A}"
12
13 win_devspeakerpc="@device_cm_{33D9A762-90C8-11D0-BD43-00A0C911CE86}\\wave_{4B0B786F-15F0-4233-818D-FFC7077ACDED}"
14 win_devmicroCasp="@device_cm_{33D9A762-90C8-11D0-BD43-00A0C911CE86}\\wave_{A53D5F12-6864-4A6C-B6E1-D21EC139A79B}"
15 # You must determine what is the aise audio header input for me hw:2,0 and the same for the native speakers
16 lin_devmicroCasp="hw:2,0"
17 lin_devspeakerpc="TODO"
18 win_list_audio=None;microCasp;speakerpc;microCasp;
19 lin_list_audio=None;microCasp;
20
21 #Fin des lignes à modifier
22

```

The lines to be edited are highlighted in yellow, we are only interested in the Windows part.

#### 3.4.1 Microphone case and built-in speakers

In this case, it will be necessary to find the values **win\_devspeakerpc** et **win\_devmicropc** .

For that, go to the directory \opt\myscreenrecorder\script

Launch **.\searchAudioDevices.cmd** we get this type of output:

```

D:\opt\myscreenrecorder\script>D:\opt\ffmpeg-latest-win64-static\bin\ffmpeg -list_devices true -f dshow -i dummy
ffmpeg version N-93580-g036b4b0f85 Copyright (c) 2000-2019 the FFmpeg developers
  built with gcc 8.3.1 (GCC) 20190414
  configuration: --enable-gpl --enable-version3 --enable-sdl2 --enable-fontconfig --enable-gnutls --enable-iconv --enable-libass --enable-libdav1d --
enable-libbluray --enable-libfreetype --enable-libmp3lame --enable-libopencore-amrnb --enable-libopencore-amrwb --enable-libopenjpeg --enable-libopus
--enable-libshine --enable-libsnappy --enable-libsoxr --enable-libtheora --enable-libtwolame --enable-libvpx --enable-libwavpack --enable-libwebp --
enable-libx264 --enable-libx265 --enable-libxml2 --enable-libzimg --enable-lzma --enable-zlib --enable-gmp --enable-libvidstab --enable-libvorbis --e
nable-libvo-amrwbenc --enable-libmysofa --enable-lspspeex --enable-libxvid --enable-libaom --enable-libmfx --enable-amf --enable-ffnvcodec --enable-c
uvid --enable-d3d11va --enable-nvenc --enable-nvdec --enable-dxva2 --enable-avisynth --enable-libopenmpt
                                libavutil      56. 26.100 / 56. 26.100
                                libavformat    58. 27.102 / 58. 27.102
                                libavcodec     58. 51.100 / 58. 51.100
                                libavdevice    58.  7.100 / 58.  7.100
                                libavfilter     7. 48.100 / 7. 48.100
                                libswscale     5.  4.100 / 5.  4.100
                                libswresample  3.  4.100 / 3.  4.100
                                libpostproc    55.  4.100 / 55.  4.100
[dshow @ 0000029da7388f00] DirectShow video devices (some may be both video and audio devices)
[dshow @ 0000029da7388f00] "HD WebCam"
[dshow @ 0000029da7388f00] Alternative name "device_cm_{33D9A762-90C8-11D0-BD43-00A0C911CE86}\wave_{4B0B786F-15F0-4233-818D-FFC7077ACDED}"
Global!
[dshow @ 0000029da7388f00] DirectShow audio devices
[dshow @ 0000029da7388f00] "Mixage st[er]eo (Realtek High Definition Audio)"
[dshow @ 0000029da7388f00] Alternative name "device_cm_{33D9A762-90C8-11D0-BD43-00A0C911CE86}\wave_{4B0B786F-15F0-4233-818D-FFC7077ACDED}"
[dshow @ 0000029da7388f00] "R[es]eau de microphones (Realtek High Definition Audio)"
[dshow @ 0000029da7388f00] Alternative name "device_cm_{33D9A762-90C8-11D0-BD43-00A0C911CE86}\wave_{A53D5F12-6864-4A6C-B6E1-D21EC139A79B}"
dummy: Immediate exit requested
D:\opt\myscreenrecorder\script>

```

Note for Linux, the command to launch is `arecord -l`, and the interface is of type `hw: 1,0` (1 and 0 to adapt according to the rank of the card and device , see <https://trac.ffmpeg.org/wiki/Capture/ALSA> )

What will interest us are the audio interfaces circled in red:

In our case we will have to fill the value of the **alternative name to avoid accented character problems**.

Speakers:

`win_devspeakerpc="@device_cm_{33D9A762-90C8-11D0-BD43-00A0C911CE86}\\wave_{4B0B786F-15F0-4233-818D-FFC7077ACDED}"`

Important, in our case, it will double the anti-slash before wave (and possibly elsewhere if it is present) otherwise the interface is not recognized.

PC microphone:

`win_devmicrpc="@device_cm_{33D9A762-90C8-11D0-BD43-00A0C911CE86}\\wave_{A53D5F12-6864-4A6C-B6E1-D21EC139A79B}"`

Important, in our case, it will double the anti-slash before wave (and possibly elsewhere if it is present) otherwise the interface is not recognized.

And then the list of audio devices must be:

`win_list_audio=None;speakerpc;micrpc;`

Save the modified file `\opt\myscreenrecorder\config\config.properties`

### 3.4.2 Headset microphone case

If you need a micro USB headset, we will take care to always take the same USB output.

We plug in our micro USB headset, and proceed as described in paragraph 3.5.1.

The output of the command : `.\searchAudioDevices.cmd`

is then :

```

C:\> Invoke-Command {
ffmpeg version N-93580-g036b4b0f85 Copyright (c) 2000-2019 the FFmpeg developers
  built with gcc 8.3.1 (GCC) 20190414
  configuration: --enable-gpl --enable-version3 --enable-sdl2 --enable-fontconfig --enable-gnutls --enable-iconv --enable-libass --enable-libdav1d --
enable-libluray --enable-libfreetype --enable-libmp3lame --enable-libopencore-amrnb --enable-libopencore-amrwb --enable-libopenjpeg --enable-libopus
--enable-lbshine --enable-liblsmas --enable-libsoxr --enable-libtheora --enable-libtwolame --enable-libvpx --enable-libwavpack --enable-libwebp --e
enable-libx264 --enable-libx265 --enable-libxml2 --enable-libz --enable-lzma --enable-zlib --enable-gmp --enable-libvidstab --enable-libvorbis --e
enable-libvo-amrwbenc --enable-libmysofa --enable-libspeex --enable-libvid --enable-libaom --enable-libbfx --enable-amf --enable-ffnvcodec --enable-c
uvid --enable-d3d11va --enable-nvenc --enable-nvdec --enable-dxva2 --enable-avisynth --enable-libopenmpt

  libavutil      56. 26.100 / 56. 26.100
  libavcodec     58. 51.100 / 58. 51.100
  libavformat    58. 27.102 / 58. 27.102
  libavdevice    58.  7.100 / 58.  7.100
  libavfilter     7. 48.100 /  7. 48.100
  libswscale      5.  4.100 /  5.  4.100
  libswresample  3.  4.100 /  3.  4.100
  libpostproc   55.  4.100 / 55.  4.100

[dshow @ 000002343eea8f00] DirectShow video devices (some may be both video and audio devices)
[dshow @ 000002343eea8f00] "HD Webcam"
[dshow @ 000002343eea8f00] Alternative name "@device_pnp_\\?\\usb\\vid_0bda&pid_5621&mi_00#6&199da46c&0&0000#{65e8773d-8f56-11d0-a3b9-00a0c9223196}
\\global"
[dshow @ 000002343eea8f00] DirectShow audio devices
[dshow @ 000002343eea8f00] "Microphone (Plantronics .Audio 400 DSP)"
[dshow @ 000002343eea8f00] Alternative name "@device_cm_{33D9A762-90C8-11D0-BD43-00A0C911CE86}\\wave_{0764C06E-1C55-44E7-B6F1-0D138DEA352A}"
[dshow @ 000002343eea8f00] Alternative name "@device_cm_{33D9A762-90C8-11D0-BD43-00A0C911CE86}\\wave_{4B08786F-15F0-4233-818D-FFC0777AC0ED}"
[dshow @ 000002343eea8f00] "Realtek High Definition Audio"
[dshow @ 000002343eea8f00] Alternative name "@device_cm_{33D9A762-90C8-11D0-BD43-00A0C911CE86}\\wave_{A53D5F12-6864-4A6C-B6E1-D21EC139A79B}"

dummy: Immediate exit requested

D:\opt\myscreenrecorder\script>

```

For my Plantronics headset microphone, the microphone is identified and must be set :

```
win_devmicrocasque="@device_cm_{33D9A762-90C8-11D0-BD43-00A0C911CE86}\\wave_{0764C06E-1C55-44E7-B6F1-0D138DEA352A}"
```

Important, in our case, it will double the anti-slash before wave (and possibly elsewhere if it is present) otherwise the interface is not recognized.

And then the list of audio devices must be:

```
win_list_audio=None;microcasque;speakerpc;micropc;
```

Save the modified file `\opt\myscreenrecorder\config\config.properties`

And “voilà” **myscreenrecorder** is ready to run.

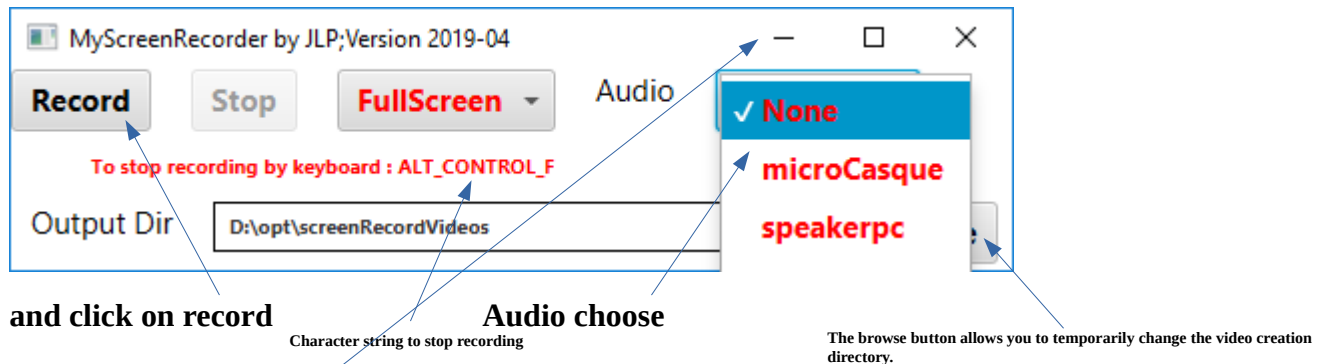
## Good use ...



# User Guide

## 4 Full screen recording

Click on the desktop icon myscreenrecorder then choose your audio output or None



and click on record

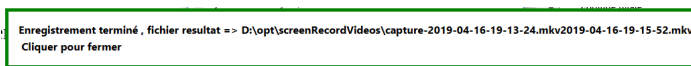
Character string to stop recording

Audio choose

The browse button allows you to temporarily change the video creation directory.

then reduce the window .

To stop, either type in the end-of-record character sequence, or re-call the window in the taskbar and click stop .



The popup shows you the location of the file results that can be seen in our favorite video software (Windows Media Player, VLC, ...).

If you want to change the video output directory permanently, you have to change the outputDir variable in the script `\opt\myscreenrecorder\script\myscreenrecorder.cmd`

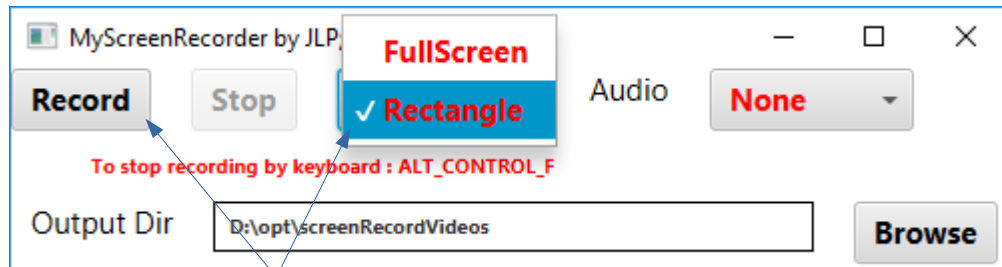
Change the highlighted line to yellow.

```
1  if exist D:\opt\ (set Disk=D) ELSE (set Disk=C)
2  Set PROJECT_HOME=%Disk%:\opt\myscreenrecorder
3  Set outputDir=%Disk%:\opt\screenRecordVideos
4  Set JAVA_HOME=%Disk%:\opt\jdk-11.0.2
5  Set JFX_HOME=%Disk%:\opt\javafx-sdk-11.0.2
6  Set pathffmpeg=%Disk%:\opt\ffmpeg-latest-win64-static
7  Set CLASSPATH=%PROJECT_HOME%\lib\jnativehook-2.1.0.jar,
```

The video can then be reworked in video editing software such as OpenShot, ShotCut or Kdenlive or other commercial software.

## 5 Recording rectangular area

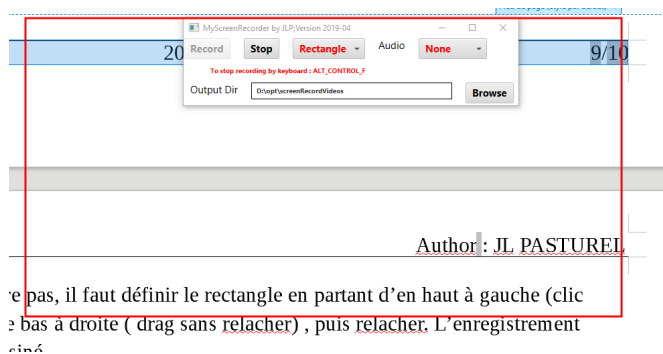
This requires an additional drawing operation of the rectangle to be recorded by drag and drop.



We choose the rectangle, and then as above we can choose the audio interface.

And then we click on record.

The recording does not start, it is necessary to define the rectangle starting from top on the left (left click without releasing) towards the bottom on the right (drag without releasing), then release. The recording starts on the drawn rectangle.



é pas, il faut définir le rectangle en partant d'en haut à gauche (clic  
e bas à droite ( drag sans relacher) , puis relacher. L'enregistrement  
ciné

At the end of the recording, it is also indicated the location of the result file.