Beginner yt-dlp Guide to Download YouTube Videos …

yt-dlp for Dummies | Download YouTube Videos / Channels Stupidly Easy

Email / Help: teoncurt106@gmail.com

This also works for livestreams, and **many** other sites than Youtube.

The key benefits of yt-dlp are the compatibility with massive amounts of videos, and wild amount of customization; making it superior to all other downloaders, but with a mild learning curve. I think it's worth it for average people (with how quick/simple it is to use), and especially archivists.

To archive streams 24/7 follow this guide, even simpler than this (there's also a section later in this document): Easy 24/7 Stream Archive Script

To rapidly clip videos just by tapping a video player, follow this guide: Clipping Made Easy

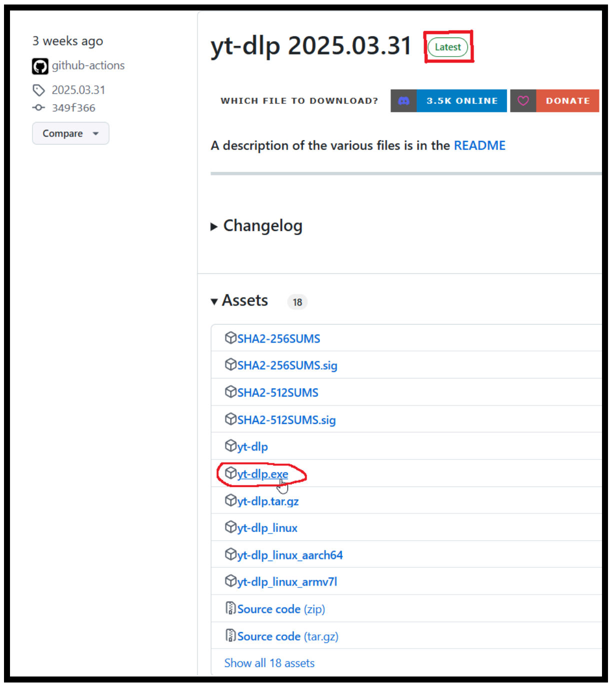
A full video tutorial of this document is being considered. I did try to make this document simple, so you shouldn't need it necessarily.

The following guide is for Windows PCs.

1. Download the NEWEST VERSION of yt-dlp.exe from this link.

   Don't run it, **it's not an actual installer**. **It runs inside another program.**

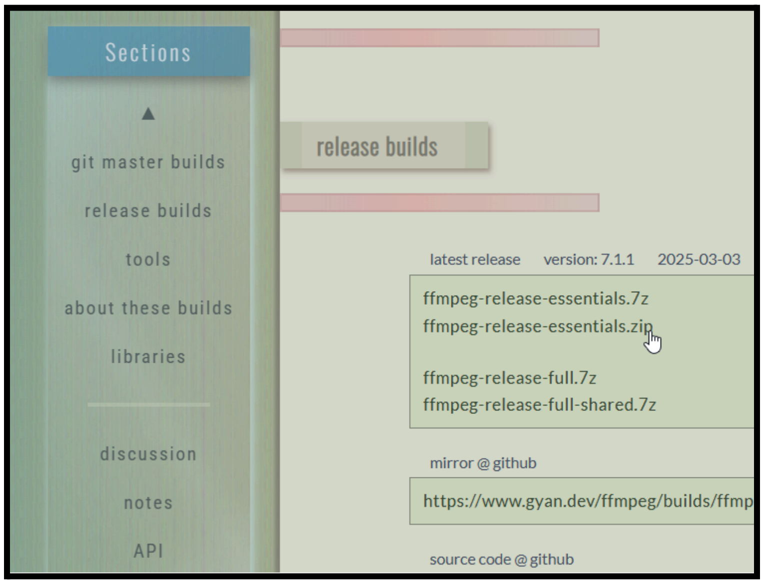
   https://github.com/yt-dlp/yt-dlp/releases.



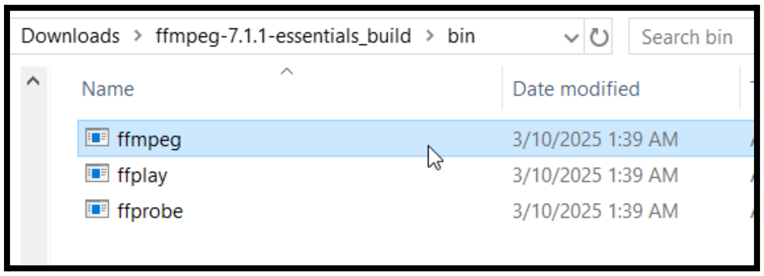
A description of the various files is in the README

2. Move it to the downloads folder, or any folder you want. *(Using downloads is my personal default)*

3. Download ffmpeg-git-essentials from this link: https://www.gyan.dev/ffmpeg/builds/.


Sections
git master builds
release builds
tools
about these builds
libraries
discussion
notes
API
release builds
latest release version: 7.1.1 2025-03-03
ffmpeg-release-essentials.7z
ffmpeg-release-essentials.zip
ffmpeg-release-full.7z
ffmpeg-release-full-shared.7z
mirror @ github
https://www.gyan.dev/ffmpeg/builds/ffmp
source code @ github

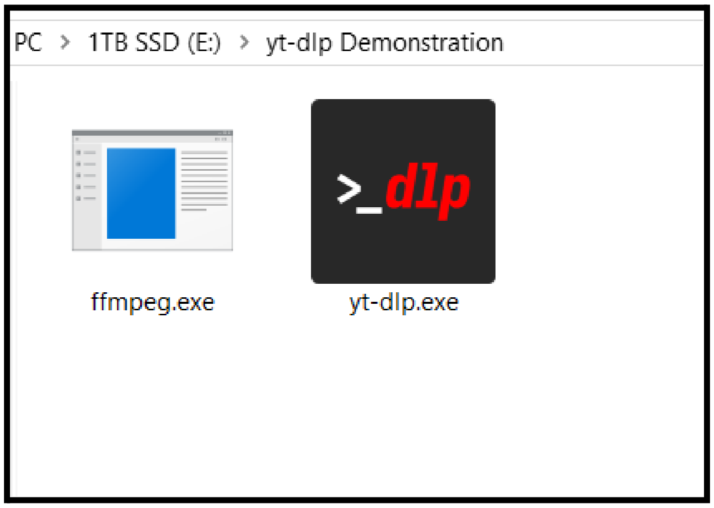
Scroll to “Release builds” and download the .zip file.



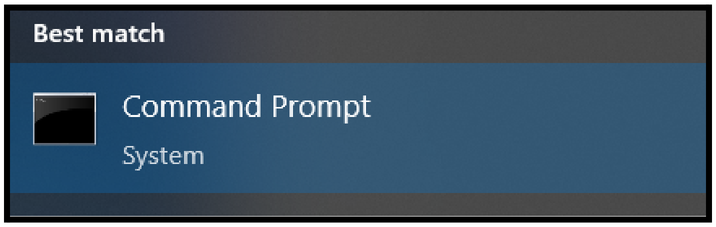
Name	Date modified
ffmpeg	3/10/2025 1:39 AM
ffplay	3/10/2025 1:39 AM
ffprobe	3/10/2025 1:39 AM

Extract to a windows folder and locate the “bin” directory.

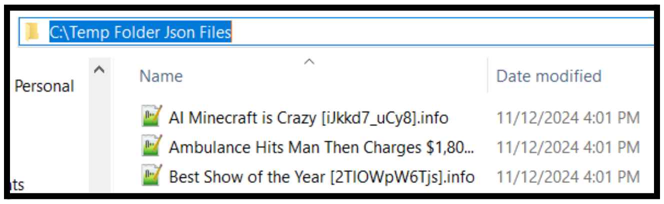
4. Get ffmpeg.exe from the bin folder (don't run it), move to downloads folder or whichever you choose. **FFmpeg is necessary for merging videos, and downloading music, so it's highly recommended to have around**.



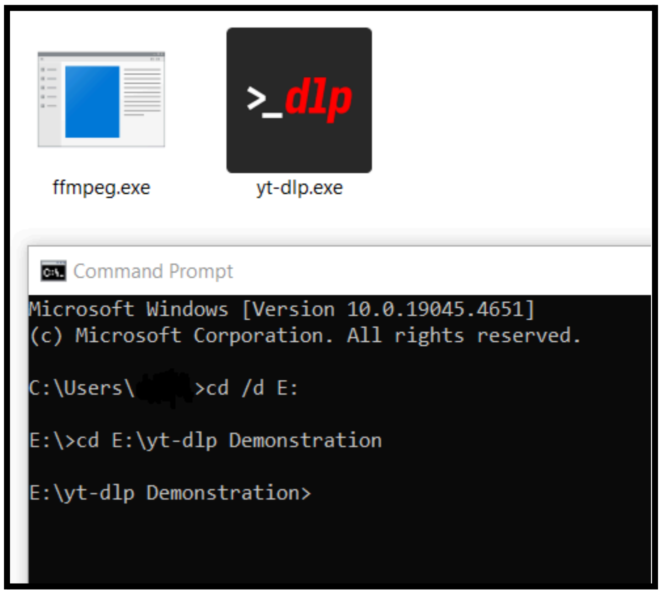
5. Open command prompt (“cmd” in the search bar).



6. ***Type “cd downloads” or “cd [filepath with yt-dlp.exe and ffmpeg”, copy it from the windows explorer address bar, Click it then copy.



Name	Date modified
AI Minecraft is Crazy [iJkkd7_uCy8].info	11/12/2024 4:01 PM
Ambulance Hits Man Then Charges $1,80...	11/12/2024 4:01 PM
Best Show of the Year [2TIOWpW6Tjs].info	11/12/2024 4:01 PM



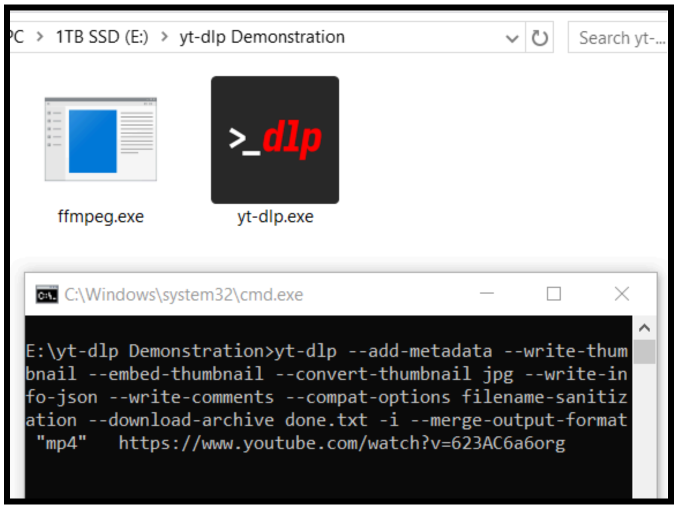
```
Microsoft Windows [Version 10.0.19045.4651]
(c) Microsoft Corporation. All rights reserved.

C:\Users\[redacted]>cd /d E:

E:\>cd E:\yt-dlp Demonstration

E:\yt-dlp Demonstration>
```

(if you need to navigate to another drive other than “C:”, you have to use this command first: cd /d [DRIVE LETTER]:)

(then cd [folder in that directory] until you get to the right folder)

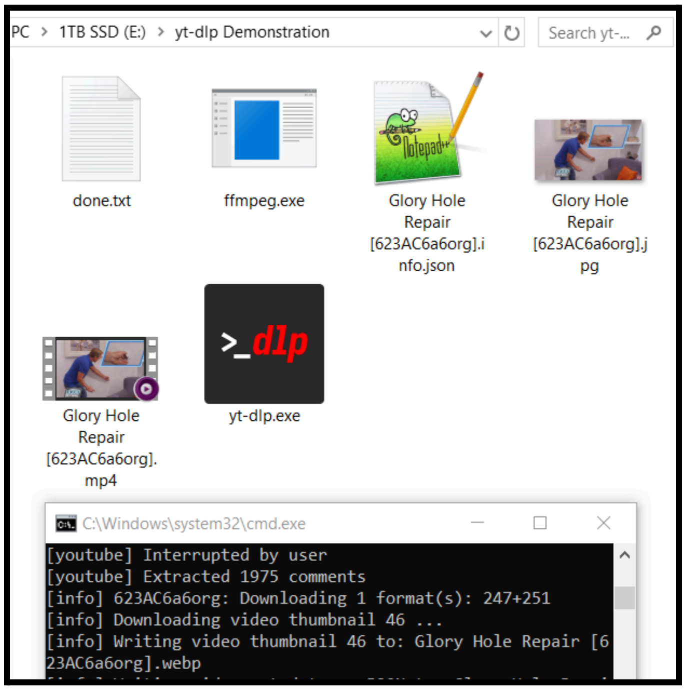
IF THIS WORKED: just copy the command and paste a URL:

yt-dlp --add-metadata -*-write-thumbnail* --embed-thumbnail --convert-thumbnail jpg -*-write-info-json* -*-write-comments* --compat-options filename-sanitization --download-archive done.txt -i --merge-output-format "mp4" -f "bv+ba" **[INSERT URL(s) HERE, or channel URL(s)]**



PC > 1TB SSD (E:) > yt-dlp Demonstration
Search yt-...
ffmpeg.exe
yt-dlp.exe
C:\Windows\system32\cmd.exe

```
E:\yt-dlp Demonstration>yt-dlp --add-metadata --write-thum
bnail --embed-thumbnail --convert-thumbnail jpg --write-in
fo-json --write-comments --compat-options filename-sanitiz
ation --download-archive done.txt -i --merge-output-format
 "mp4"   https://www.youtube.com/watch?v=623AC6a6org
```



PC > 1TB SSD (E:) > yt-dlp Demonstration
Search yt-...
done.txt
ffmpeg.exe
Glory Hole Repair [623AC6a6org].info.json
Glory Hole Repair [623AC6a6org].jpg
Glory Hole Repair [623AC6a6org].mp4
yt-dlp.exe
C:\Windows\system32\cmd.exe

```
[youtube] Interrupted by user
[youtube] Extracted 1975 comments
[info] 623AC6a6org: Downloading 1 format(s): 247+251
[info] Downloading video thumbnail 46 ...
[info] Writing video thumbnail 46 to: Glory Hole Repair [6
23AC6a6org].webp
```

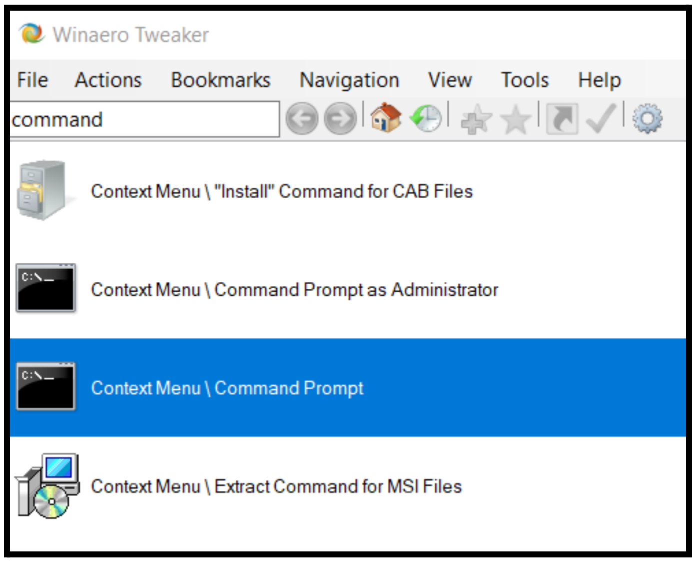
I. **IF COMMAND PROMPT NAVIGATION IS TOO WEIRD, INSTALL THIS PROGRAM:** https://winaero.com/download-winaero-tweaker/

<u>This is the best option anyway, highly recommended.</u>

It makes it much quicker to use the program.

All you'll have to do is right click a folder with yt-dlp.exe in it, click "Open command window here," and then paste your command.

Open the program and search "Command Prompt"

II. 

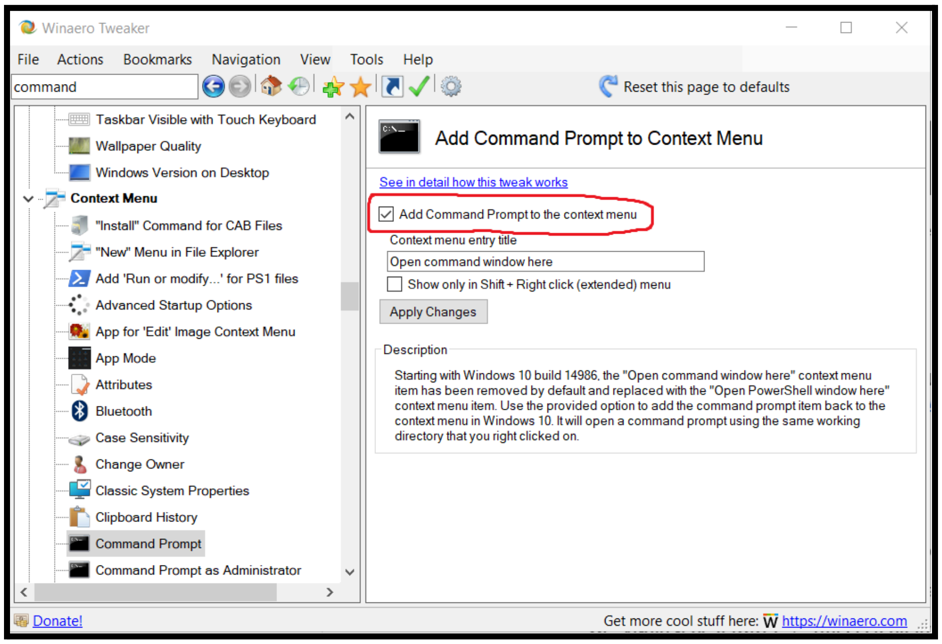
command

Context Menu \ "Install" Command for CAB Files

Context Menu \ Command Prompt as Administrator

Context Menu \ Command Prompt

Context Menu \ Extract Command for MSI Files

III. 
Winaero Tweaker
File Actions Bookmarks Navigation View Tools Help
Reset this page to defaults
Taskbar Visible with Touch Keyboard
Wallpaper Quality
Windows Version on Desktop
Context Menu
"Install" Command for CAB Files
"New" Menu in File Explorer
Add 'Run or modify...' for PS1 files
Advanced Startup Options
App for 'Edit' Image Context Menu
App Mode
Attributes
Bluetooth
Case Sensitivity
Change Owner
Classic System Properties
Clipboard History
Command Prompt
Command Prompt as Administrator
Donate!
Get more cool stuff here: https://winaero.com

command

Add Command Prompt to Context Menu

See in detail how this tweak works

[✓] Add Command Prompt to the context menu

Context menu entry title

Open command window here

[ ] Show only in Shift + Right click (extended) menu

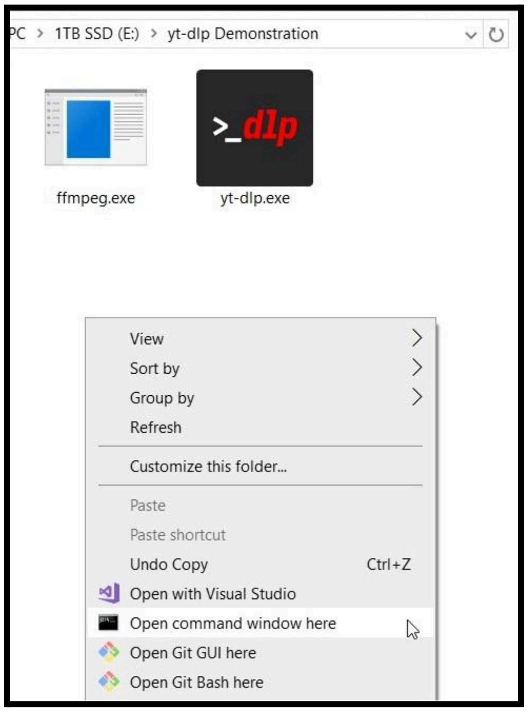
Apply Changes

Description

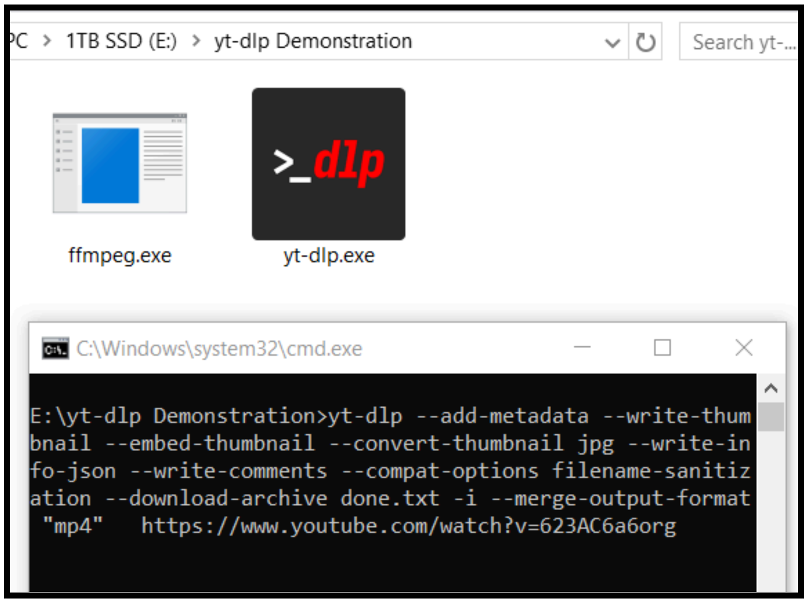
Starting with Windows 10 build 14986, the "Open command window here" context menu item has been removed by default and replaced with the "Open PowerShell window here" context menu item. Use the provided option to add the command prompt item back to the context menu in Windows 10. It will open a command prompt using the same working directory that you right clicked on.

IV. Right click in the folder you have yt-dlp.exe and ffmpeg.exe.

(scroll to the next section to learn how to download videos in any folder).

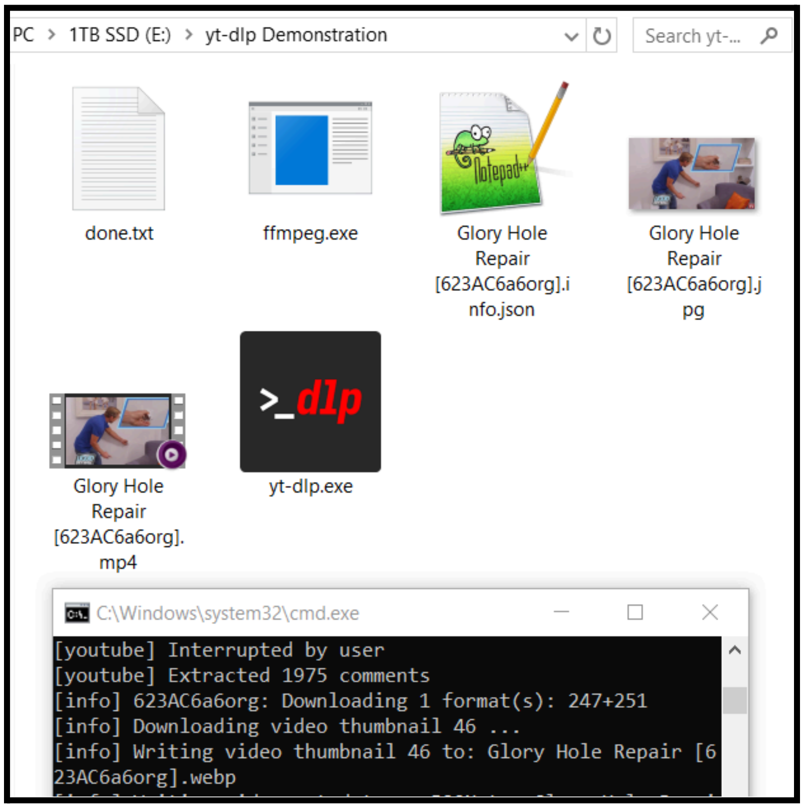


V. Paste the command, and any videos you want to download.


PC > 1TB SSD (E:) > yt-dlp Demonstration
Search yt-...
ffmpeg.exe
yt-dlp.exe
C:\Windows\system32\cmd.exe

```
E:\yt-dlp Demonstration>yt-dlp --add-metadata --write-thum
bnail --embed-thumbnail --convert-thumbnail jpg --write-in
fo-json --write-comments --compat-options filename-sanitiz
ation --download-archive done.txt -i --merge-output-format
 "mp4"   https://www.youtube.com/watch?v=623AC6a6org
```

VI. Hit enter to Download video(s).

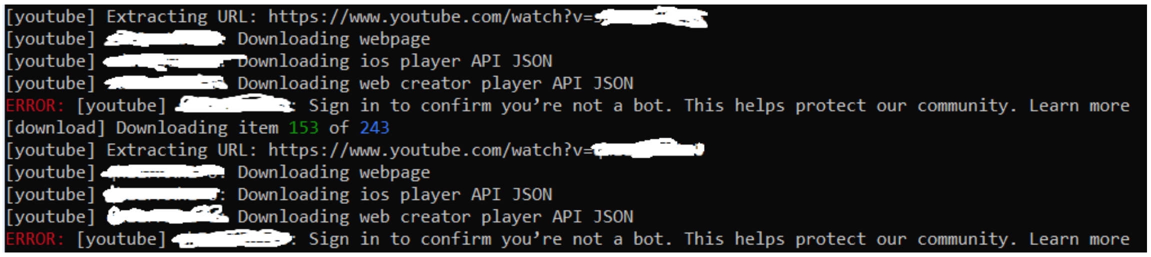

PC > 1TB SSD (E:) > yt-dlp Demonstration
Search yt-...
done.txt
ffmpeg.exe
Glory Hole Repair [623AC6a6org].info.json
Glory Hole Repair [623AC6a6org].jpg
Glory Hole Repair [623AC6a6org].mp4
yt-dlp.exe
C:\Windows\system32\cmd.exe

```
[youtube] Interrupted by user
[youtube] Extracted 1975 comments
[info] 623AC6a6org: Downloading 1 format(s): 247+251
[info] Downloading video thumbnail 46 ...
[info] Writing video thumbnail 46 to: Glory Hole Repair [6
23AC6a6org].webp
```

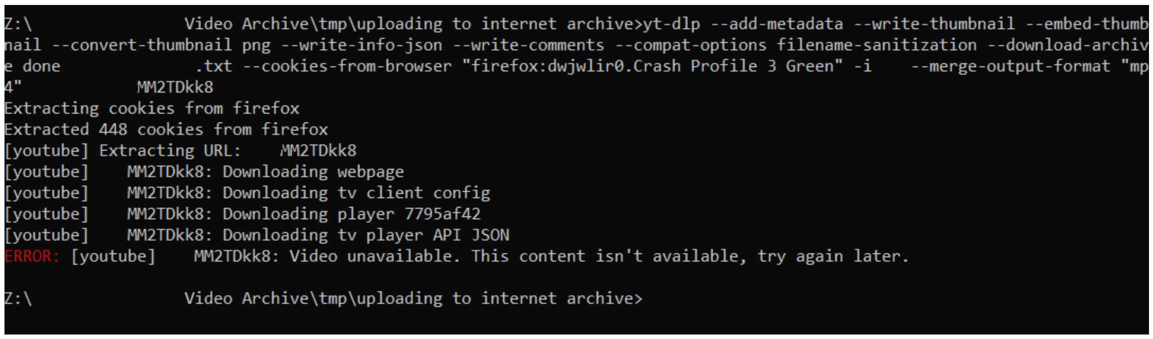
Now you can type "*yt-dlp [INSERT_URL]*" to download a video, or an entire Youtube channel/playlist (at max quality). Scroll down for my command templates and important tips.

IMPORTANT: if you download a large playlist or channel, you will likely run into certain errors which are a result of youtube attempting to impede video downloading programs to cement their monopoly (starting in 2024), getting worse in 2025.

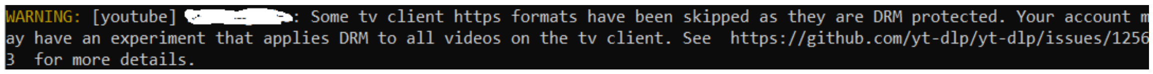
This is almost certainly due to AI video companies scraping videos to train their models. Remember, Youtube wants the ability to scrape ***their own*** videos for ***their*** AI models.



```
[youtube] Extracting URL: https://www.youtube.com/watch?v=[redacted]
[youtube] [redacted] Downloading webpage
[youtube] [redacted]Downloading ios player API JSON
[youtube] [redacted] Downloading web creator player API JSON
ERROR: [youtube] [redacted]: Sign in to confirm you’re not a bot. This helps protect our community. Learn more
[download] Downloading item 153 of 243
[youtube] Extracting URL: https://www.youtube.com/watch?v=[redacted]
[youtube] [redacted]: Downloading webpage
[youtube] [redacted]: Downloading ios player API JSON
[youtube] [redacted] Downloading web creator player API JSON
ERROR: [youtube] [redacted]: Sign in to confirm you’re not a bot. This helps protect our community. Learn more
```



```
Z:\               Video Archive\tmp\uploading to internet archive>yt-dlp --add-metadata --write-thumbnail --embed-thumb
nail --convert-thumbnail png --write-info-json --write-comments --compat-options filename-sanitization --download-archiv
e done              .txt --cookies-from-browser "firefox:dwjwlir0.Crash Profile 3 Green" -i    --merge-output-format "mp
4"            MM2TDkk8
Extracting cookies from firefox
Extracted 448 cookies from firefox
[youtube] Extracting URL:    MM2TDkk8
[youtube]    MM2TDkk8: Downloading webpage
[youtube]    MM2TDkk8: Downloading tv client config
[youtube]    MM2TDkk8: Downloading player 7795af42
[youtube]    MM2TDkk8: Downloading tv player API JSON
ERROR: [youtube]    MM2TDkk8: Video unavailable. This content isn't available, try again later.

Z:\               Video Archive\tmp\uploading to internet archive>
```



```
WARNING: [youtube] [redacted]: Some tv client https formats have been skipped as they are DRM protected. Your account m
ay have an experiment that applies DRM to all videos on the tv client. See  https://github.com/yt-dlp/yt-dlp/issues/1256
3  for more details.
```

I have several techniques you can try to eliminate or reduce the impact of these errors, scroll to “DISCLAIMER FOR VPN USERS” section; it doesn’t just apply to VPNs.

It’s not that hard to combat, so I wouldn’t be discouraged. This is mandatory now with Google being the equivalent of satan.

****Note:* command prompt starts out from this folder “C:\Users\[USER]”

If your folder was located inside the Documents folder, you’d type these commands in order, then you can run the command.

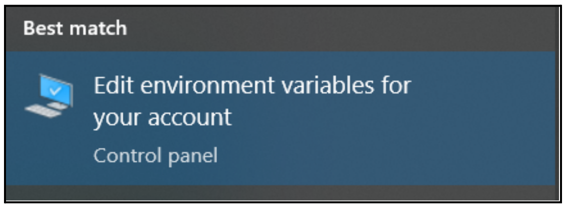
cd documents	(click enter)
cd [YOUR FOLDER]	(click enter)

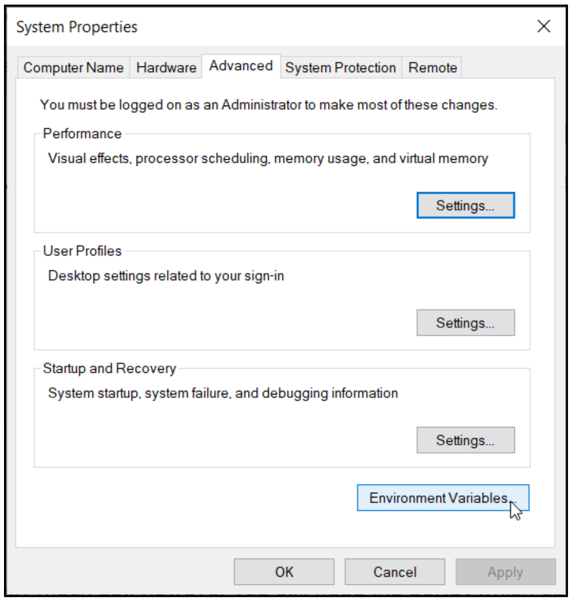
Set yt-dlp/ffmpeg to be Accessible Anywhere

You can also set yt-dlp.exe and FFmpeg.exe in your PATH, so it can be referenced anywhere from your PC.

<u>**You won't have to have yt-dlp.exe or ffmpeg.exe in the folder you'll use it.**</u>

Like the WinAero context menu button, <u>this will make your life easier</u> so you should just do it.




System Properties

Computer Name | Hardware | Advanced | System Protection | Remote

You must be logged on as an Administrator to make most of these changes.

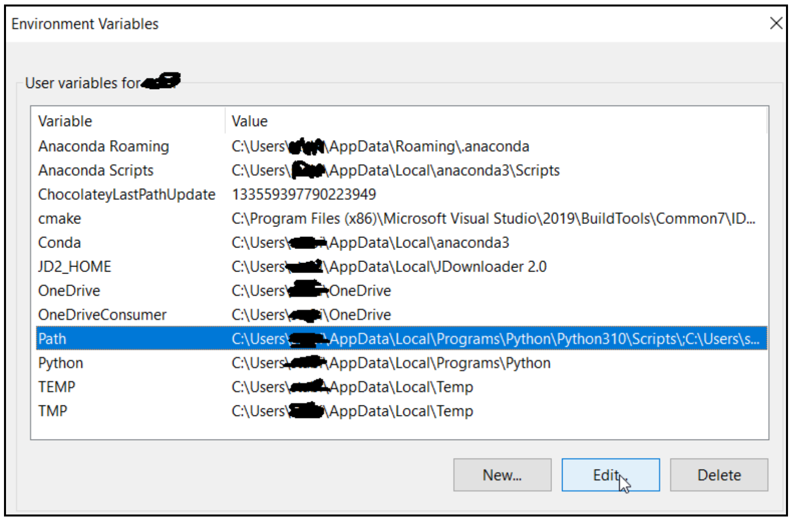
Performance
Visual effects, processor scheduling, memory usage, and virtual memory
Settings...

User Profiles
Desktop settings related to your sign-in
Settings...

Startup and Recovery
System startup, system failure, and debugging information
Settings...

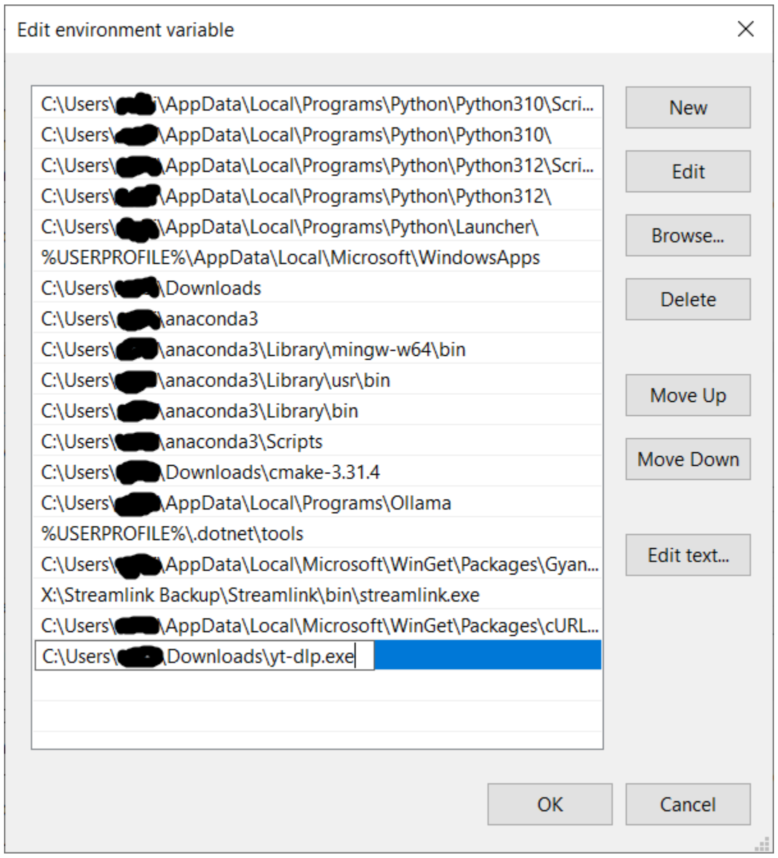
Environment Variables...

OK Cancel Apply



Variable	Value
Anaconda Roaming	C:\Users\[redacted]\AppData\Roaming\.anaconda
Anaconda Scripts	C:\Users\[redacted]\AppData\Local\anaconda3\Scripts
ChocolateyLastPathUpdate	133559397790223949
cmake	C:\Program Files (x86)\Microsoft Visual Studio\2019\BuildTools\Common7\ID...
Conda	C:\Users\[redacted]\AppData\Local\anaconda3
JD2_HOME	C:\Users\[redacted]\AppData\Local\JDownloader 2.0
OneDrive	C:\Users\[redacted]\OneDrive
OneDriveConsumer	C:\Users\[redacted]\OneDrive
Path	C:\Users\[redacted]\AppData\Local\Programs\Python\Python310\Scripts\;C:\Users\s...
Python	C:\Users\[redacted]\AppData\Local\Programs\Python
TEMP	C:\Users\[redacted]\AppData\Local\Temp
TMP	C:\Users\[redacted]\AppData\Local\Temp

Set the full directory where yt-dlp is located as a new PATH variable (which you will continually update as needed).



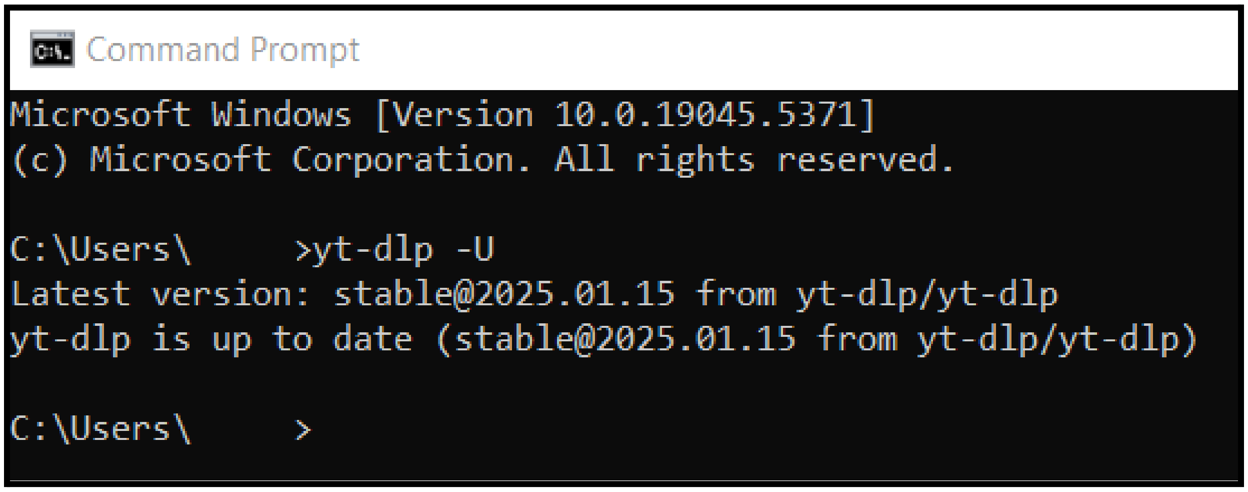
```
C:\Users\[redacted]\AppData\Local\Programs\Python\Python310\Scri...
C:\Users\[redacted]\AppData\Local\Programs\Python\Python310\
C:\Users\[redacted]\AppData\Local\Programs\Python\Python312\Scri...
C:\Users\[redacted]\AppData\Local\Programs\Python\Python312\
C:\Users\[redacted]\AppData\Local\Programs\Python\Launcher\
%USERPROFILE%\AppData\Local\Microsoft\WindowsApps
C:\Users\[redacted]\Downloads
C:\Users\[redacted]\anaconda3
C:\Users\[redacted]\anaconda3\Library\mingw-w64\bin
C:\Users\[redacted]\anaconda3\Library\usr\bin
C:\Users\[redacted]\anaconda3\Library\bin
C:\Users\[redacted]\anaconda3\Scripts
C:\Users\[redacted]\Downloads\cmake-3.31.4
C:\Users\[redacted]\AppData\Local\Programs\Ollama
%USERPROFILE%\.dotnet\tools
C:\Users\[redacted]\AppData\Local\Microsoft\WinGet\Packages\Gyan...
X:\Streamlink Backup\Streamlink\bin\streamlink.exe
C:\Users\[redacted]\AppData\Local\Microsoft\WinGet\Packages\cURL...
C:\Users\[redacted]\Downloads\yt-dlp.exe
```

KEEP YT-DLP UPDATED OR IT WILL BREAK!

This is mandatory, or yt-dlp **will** stop working and give 403 Forbidden / nsig extraction errors after about a month or even a few days, because Youtube continues to fight the fruitless war against Adblockers. yt-dlp will not stand down either.

All you have to do is run this command in the folder where you have yt-dlp.exe, the same way you'd run any other command.

yt-dlp -U



```
Microsoft Windows [Version 10.0.19045.5371]
(c) Microsoft Corporation. All rights reserved.

C:\Users\     >yt-dlp -U
Latest version: stable@2025.01.15 from yt-dlp/yt-dlp
yt-dlp is up to date (stable@2025.01.15 from yt-dlp/yt-dlp)

C:\Users\     >
```

Don't be surprised to see yt-dlp updated just a few days ago, and then stop working with an error like 403 or nsig: this is regular at this point.

It gets updated a few times a month sometimes, and you'll have to do this as well to keep using it.
You can run yt-dlp Nightly Builds to be most up-to-date without a 100% guarantee of functionality; this is generally recommended now.

yt-dlp --update-to nightly

Command I use for videos

It includes thumbnails and .jsons with comments for metadata (feel free to remove those commands). Just copy and paste the url anywhere after the command.

Note: If you want to include the .json files for video metadata and Youtube comments, make sure to use a VPN, or don't publish the files publicly because they include your IP address.

You can also run this python script to instantly clear out IPs in a directory of .jsons. If you need to quickly learn how to run scripts from Python IDLE; refer to the "SPECIAL TOOLS" section for a guide. It's quite easy even for non-coders, like myself.

cd downloads

yt-dlp --add-metadata *--write-thumbnail* --embed-thumbnail --convert-thumbnail jpg *--write-info-json --write-comments* --compat-options filename-sanitization --download-archive done.txt -i --merge-output-format "mp4" -f "bv+ba"   **[URL(s) GO HERE]**

- **If you want your videos to have the channel name in front of the title, add this command:**

  -o "%(uploader)s - %(title)s [%(id)s]"

  Format: [Channel Name] – [Video Title] [URL ID]

- **If you want your videos to have the upload date in front of the title, add this command:**

  -o "[%(upload_date>%Y-%m-%d)s] %(title)s [%(id)s]"

Format: [YYYY-MM-DD] [Video Title] [URL ID]

Command I use for music

(in .mp3 format or whichever you prefer)

(.flac is an ideal format because it’s uncompressed, but it’s not as widely supported)

```
cd downloads
```

```
yt-dlp --prefer-ffmpeg --extract-audio --audio-format mp3 --add-metadata
--write-info-json --write-comments --write-thumbnail --embed-thumbnail
--convert-thumbnail jpg --compat-options filename-sanitization -i
--download-archive donemp3.txt -o "%(uploader)s - %(title)s [%(id)s]"
```
[URL(s) GO HERE]

^^^ this version of the -o command makes the file name like this: ^^^

[Channel Name] – [Video Title] [URL ID]

DISCLAIMER FOR VPN USERS / If you get “Sign in” Errors

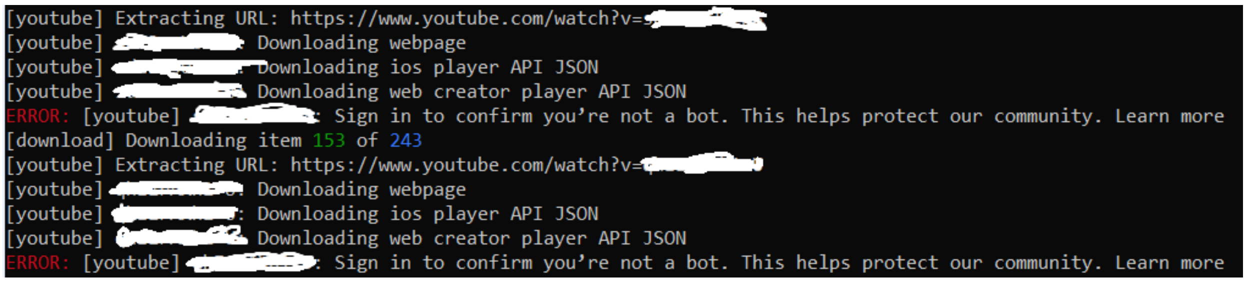
Without getting into personal opinion, Google and Youtube are getting more and more authoritarian when it comes to (mostly) harmless VPN users (not to mention, adblockers in general). People simply wanting to watch geo-restricted content in their country could now be outright blocked by Google, because of attempts to block bots.

Even one of the most common and well-respected VPNs, Mullvad, can:

- Get you labeled as a bot when google searching, requiring a captcha that might not even allow you to finish it.
- Block attempts to login to Gmail accounts without phone verification (which won’t happen once you turn it off). **This can be solved easily with WinAuth authentication codes, and you can enable it by turning on 2-FA in Google (you don’t even need a phone # to set it up; there’s a skip option).** Make sure to make backup codes. Good idea with or without Mullvad.
- Block most attempts at making Gmail accounts without phone verification.
- Force “Verify it's You” popups within Youtube Studio, which show up and block you from viewing your channel. It might force you to go through long verification processes likely requiring a phone number, or just blocking you entirely. **Again, easy to solve using WinAuth.**

- **Show “Confirm you’re not a bot” when opening Youtube videos in a Private/Incognito window.**

<u>This final point is especially relevant</u>, because it now applies to many yt-dlp users who use VPNs, and download more than a couple videos, which is kind of the point in many cases.



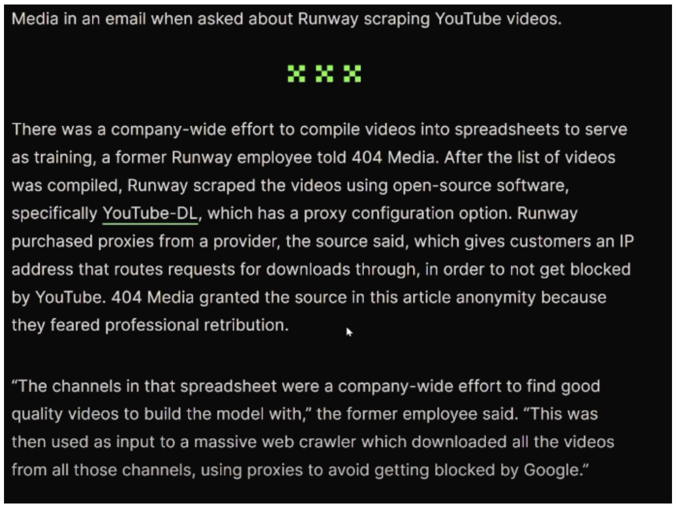
```
[youtube] Extracting URL: https://www.youtube.com/watch?v=[redacted]
[youtube] [redacted] Downloading webpage
[youtube] [redacted] Downloading ios player API JSON
[youtube] [redacted] Downloading web creator player API JSON
ERROR: [youtube] [redacted]: Sign in to confirm you’re not a bot. This helps protect our community. Learn more
[download] Downloading item 153 of 243
[youtube] Extracting URL: https://www.youtube.com/watch?v=[redacted]
[youtube] [redacted] Downloading webpage
[youtube] [redacted]: Downloading ios player API JSON
[youtube] [redacted] Downloading web creator player API JSON
ERROR: [youtube] [redacted]: Sign in to confirm you’re not a bot. This helps protect our community. Learn more
```

If you want to download large amounts of videos with yt-dlp, there’s now a fairly high chance you will see this exact error after some period of time. It may be within minutes, hours, days but it’ll probably happen.

This can even happen WITHOUT a VPN. Yes, Youtube is now blocking home IPs that are downloading large amounts of content with yt-dlp.

I theorize this could be related to RunwayML's usage of yt-dlp + proxies when they scraped thousands of youtube videos to train their AI video models, which was exposed in July 2024. VPNs are a form of proxies. Just speculation. Fuck RunwayML.

These AI companies are probably using yt-dlp to do this in mass, and youtube knows.



Media in an email when asked about Runway scraping YouTube videos.

There was a company-wide effort to compile videos into spreadsheets to serve as training, a former Runway employee told 404 Media. After the list of videos was compiled, Runway scraped the videos using open-source software, specifically YouTube-DL, which has a proxy configuration option. Runway purchased proxies from a provider, the source said, which gives customers an IP address that routes requests for downloads through, in order to not get blocked by YouTube. 404 Media granted the source in this article anonymity because they feared professional retribution.

"The channels in that spreadsheet were a company-wide effort to find good quality videos to build the model with," the former employee said. "This was then used as input to a massive web crawler which downloaded all the videos from all those channels, using proxies to avoid getting blocked by Google."

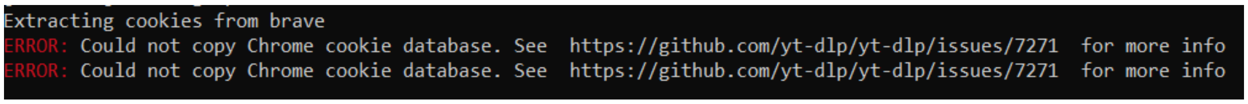
Once they blacklist your IP, if you're on a VPN you can switch servers but they can eventually blacklist your new IP, after some period of time. **You can turn your VPN off, which typically works, but:**

- **If you're privacy-minded or constantly connected through VPNs, you might not want to.** *(Though, using another DNS server is more important anyway, it's free to do and you need to do it).*
- **They can blacklist your *residential* IP eventually, which seems to wear off after a couple days but is still appalling.**
- **Turning off your VPN will reveal your IP address inside the .info.json files you download.** I have a python script that can delete all IP addresses in jsons, it's mainly a concern if you upload them anywhere public.

There are Several Strategies to Fix This.

The strategy I use and have success with is Browser Cookies + A Tab Refresher (inside Firefox).

You can try this with Chrome or Brave as well, but in my case Chrome/Brave doesn't want to send cookies to yt-dlp, so I use Firefox exclusively (for cookies).



```
Extracting cookies from brave
ERROR: Could not copy Chrome cookie database. See  https://github.com/yt-dlp/yt-dlp/issues/7271  for more info
ERROR: Could not copy Chrome cookie database. See  https://github.com/yt-dlp/yt-dlp/issues/7271  for more info
```

Add this command to your yt-dlp command; you must be signed into a Youtube account in the browser of your choosing:

--cookies-from-browser chrome

--cookies-from-browser firefox

--cookies-from-browser brave

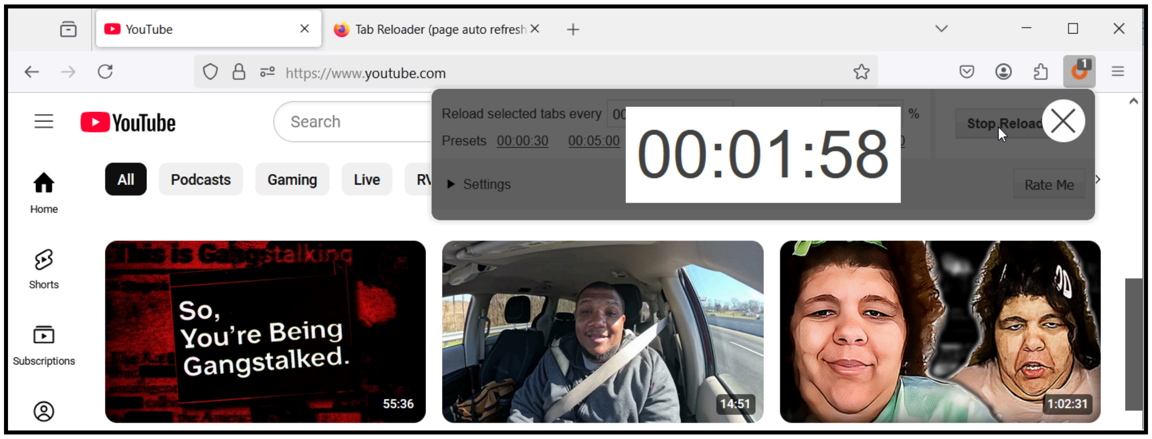
This alone probably won't work for very long, so:

1. Install this Firefox addon. It also exists on the Chrome webstore.

   https://addons.mozilla.org/en-US/firefox/addon/tab-reloader/

2. Now leave an open tab on youtube inside the browser *(LOGGED INTO YOUTUBE)*, *or specific browser profile*, that you are using in Firefox.

   Set the tab reloader to reload every 1-4 minutes depending on the magnitude of your downloads.



As long as you leave this tab open, the cookies will be refreshed and should work fine (*almost*) indefinitely.

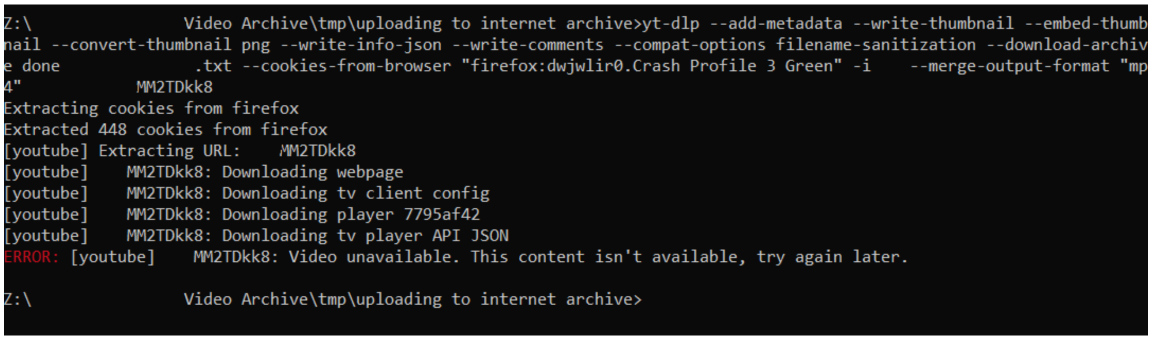
If you are mass-downloading in multiple windows, or for long periods, you may need to monitor your downloads for sign-in errors.

You can still get Sign in errors within 2 minute intervals sometimes, but if you use the "Cookie Auto-Rotation Script" you won't have to worry about this risk (scroll down for more).

<u>**Manually refreshing the tab, and re-running your command will fix the issue most of the time, but not always.**</u>

<u>At some point, you may also see this error</u>

Video unavailable. This content isn't available, try again later.



```
Z:\                Video Archive\tmp\uploading to internet archive>yt-dlp --add-metadata --write-thumbnail --embed-thumb
nail --convert-thumbnail png --write-info-json --write-comments --compat-options filename-sanitization --download-archiv
e done           .txt --cookies-from-browser "firefox:dwjwlir0.Crash Profile 3 Green" -i    --merge-output-format "mp
4"           MM2TDkk8
Extracting cookies from firefox
Extracted 448 cookies from firefox
[youtube] Extracting URL:    MM2TDkk8
[youtube]    MM2TDkk8: Downloading webpage
[youtube]    MM2TDkk8: Downloading tv client config
[youtube]    MM2TDkk8: Downloading player 7795af42
[youtube]    MM2TDkk8: Downloading tv player API JSON
ERROR: [youtube]    MM2TDkk8: Video unavailable. This content isn't available, try again later.

Z:\                Video Archive\tmp\uploading to internet archive>
```

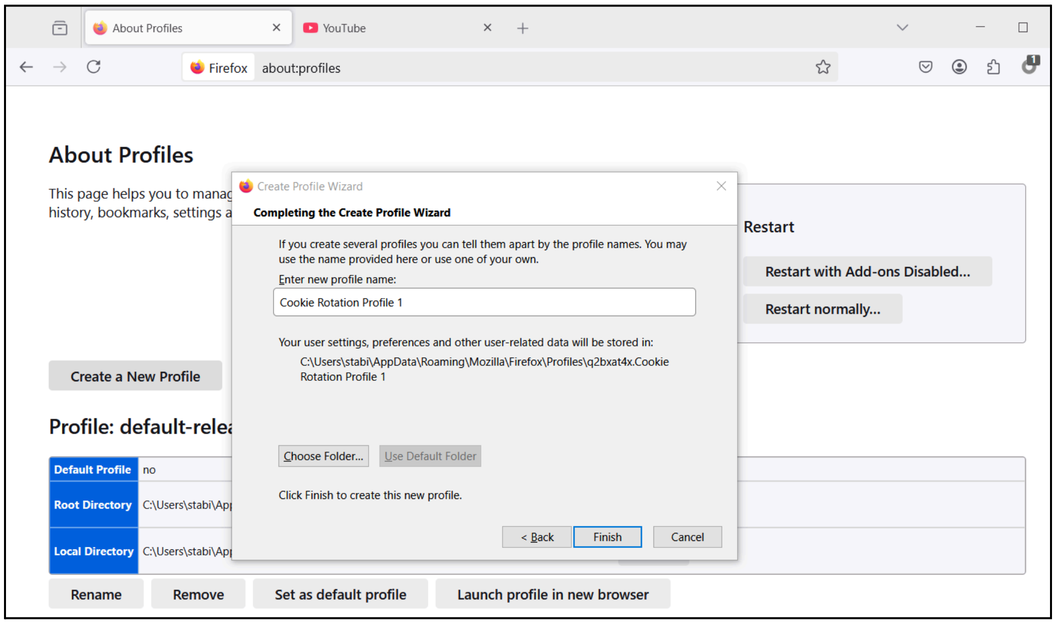
<u>This cannot be fixed by switching your IP.</u>

This *seems* to be tied to your browser profile itself, and not the browser cookies, because refreshing the cookies won't fix this error; neither will switching to another Youtube/Gmail account.

Removing the cookies command *will* prevent it from happening, but again, this won't work for long. When you're mass-downloading, this is almost a deal-breaker.

<u>The only way to fix this particular error is to switch to another browser profile WITH tab reloader. Or wait for the blacklist to expire (luckily, it does).</u>

- **In my case, I have several browser profiles open, all with tab reloader and logged into different youtube accounts. I only will open them when needed for yt-dlp.**
- When one browser profile gets blocked (presumably, permanently) I just make a new one, install the tab reloader, log into youtube, setup the refresher, and copy the firefox browser profile name from ***C:\Users\[USER]\AppData\Roaming\Mozilla\Firefox\Profiles*.**



About Profiles

This page helps you to manag... history, bookmarks, settings a...

Create Profile Wizard

Completing the Create Profile Wizard

If you create several profiles you can tell them apart by the profile names. You may use the name provided here or use one of your own.

Enter new profile name: Cookie Rotation Profile 1

Your user settings, preferences and other user-related data will be stored in:
C:\Users\stabi\AppData\Roaming\Mozilla\Firefox\Profiles\q2bxat4x.Cookie Rotation Profile 1

Click Finish to create this new profile.

Profile: default-relea...

Default Profile	no
Root Directory	C:\Users\stabi\App
Local Directory	C:\Users\stabi\App

You probably want to create at least 4 new profiles, especially if you see a "DRM protected" warning when downloading (discussed below). This may permanently block your profile.

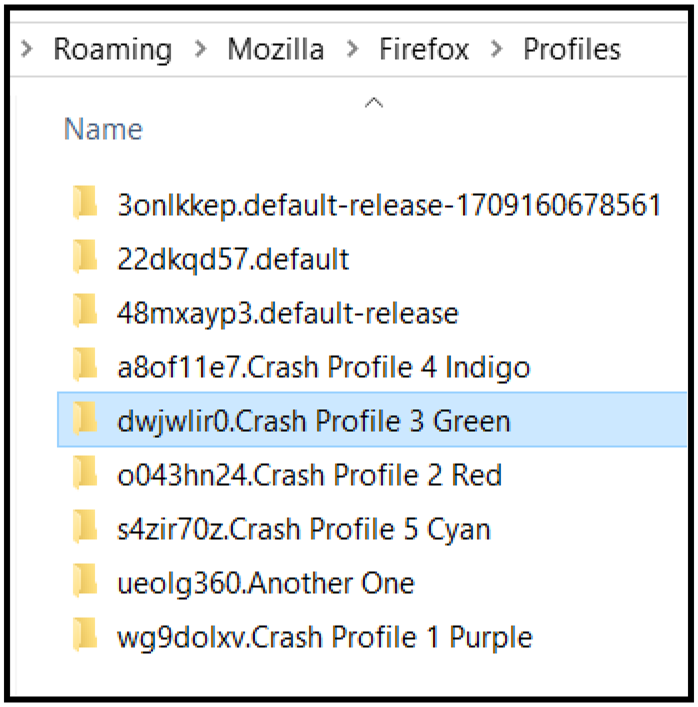
- **<u>You'll need to know how to properly identify different firefox profiles in your YT-DLP command.</u>**

- **Take for example, this command I use by default:**

  --cookies-from-browser "firefox:dwjwlir0.Crash Profile 3 Green"

<u>You have to find the precise name of the firefox profile, from your Firefox AppData/Roaming Folder</u>

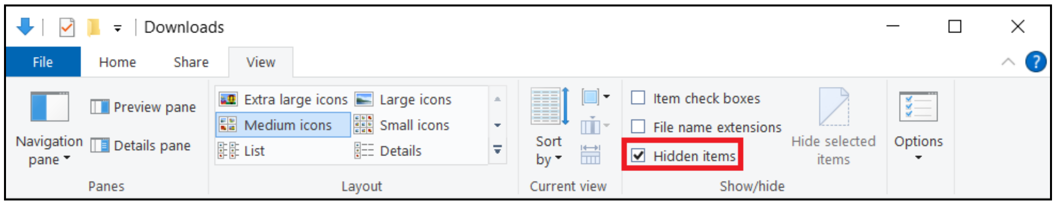
C:\Users\[USER]\AppData\Roaming\Mozilla\Firefox\Profiles


Roaming > Mozilla > Firefox > Profiles

Name

3onlkkep.default-release-1709160678561
22dkqd57.default
48mxayp3.default-release
a8of11e7.Crash Profile 4 Indigo
dwjwlir0.Crash Profile 3 Green
o043hn24.Crash Profile 2 Red
s4zir70z.Crash Profile 5 Cyan
ueolg360.Another One
wg9dolxv.Crash Profile 1 Purple

If you can't find the AppData folder inside your **C:\Users\[USER]** directory, you must enable "Hidden Items" under the View Tab in File Explorer.



<u>Take the folder name, and paste it into this command template.</u>

<u>Between the colon, and the end quotation mark.</u>

--cookies-from-browser "firefox:"

--cookies-from-browser "firefox:o043hn24.Crash Profile 2 Red"

- **If I am running a command on many channels, I <u>make sure</u> to have the –download-archive file.txt command, and I run the command in**

multiple firefox profiles, in order. You must use –download-archive file.txt.

- **Effectively, I'm running multiple yt-dlp commands on the same channel or list of URLs. The only difference is the firefox profile portion of the command.**

  I will take these commands, and paste them into Command Prompt one after another in the same window, NOT in multiple windows.

  No you don't need to wait for the first command to finish before pasting and hitting enter. Paste the first one and hit enter, paste the 2nd and so on. All of them will run one after another within command prompt.

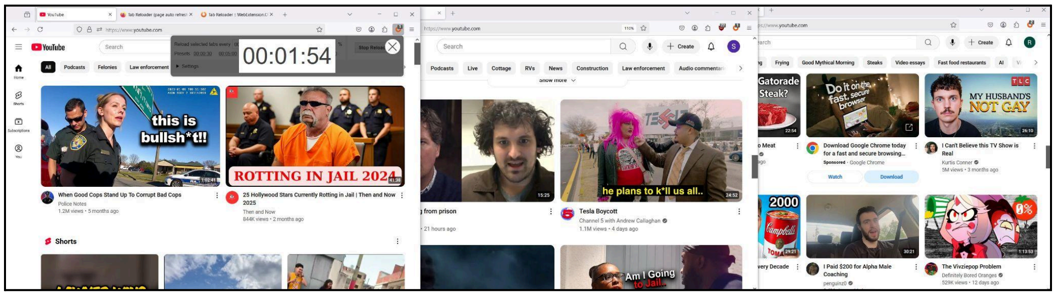
  If the final URL says "already recorded in archive" then you have all the videos in your command.

yt-dlp --add-metadata --write-thumbnail --embed-thumbnail --convert-thumbnail jpg --write-info-json --write-comments --compat-options filename-sanitization --download-archive C:/Users/[USER]/Downloads/done.txt -i --merge-output-format "mp4" **--cookies-from-browser "firefox:wg9dolxv.Crash Profile 1 Purple"** https://www.youtube.com/@BestEverFoodReviewShow

yt-dlp --add-metadata --write-thumbnail --embed-thumbnail --convert-thumbnail jpg --write-info-json --write-comments --compat-options filename-sanitization --download-archive C:/Users/[USER]/Downloads/done.txt -i --merge-output-format "mp4" **--cookies-from-browser "firefox:o043hn24.Crash Profile 2 Red"** https://www.youtube.com/@BestEverFoodReviewShow

yt-dlp --add-metadata --write-thumbnail --embed-thumbnail --convert-thumbnail jpg --write-info-json --write-comments --compat-options filename-sanitization --download-archive C:/Users/[USER]/Downloads/done.txt -i --merge-output-format "mp4" **--cookies-from-browser "firefox:dwjwlir0.Crash Profile 3 Green**" https://www.youtube.com/@BestEverFoodReviewShow

For all the profiles I'm using, I'll have a separate firefox window open, logged into youtube, with tab reloading. Minimized of course.



- **This will work because each command, in a different firefox profile, will pick up where the other one left off.**

  If the first command downloaded some and got a *video unavailable* error, other commands will continue downloading. If the first command worked perfectly, the other commands won't re-download anything.

  (Scroll to "Cookie Auto-Rotation Script" for an automated way to accomplish this).

  If you're taking mass-archival seriously, this is one of the best options to avoid being subjugated by Google blacklists.

Other Methods

- **Sleep Interval.** The simplest is to add a sleep interval to your download command. I have not had success with it but you may. Best to try this first as it's just an extra argument.

  --sleep-interval 10 --max-sleep-interval 30

- **Switch VPN server.** This is the quickest and most consistent solution (for "Sign In" errors). But if you're downloading large channels, it might get annoying having to randomly switch after checking your window and seeing if there are errors.

- **File cookies.** While browser cookies generally fix most issues, using a cookies.txt file is typically the worst option, and will expire after one or a few uses.

  - ➔ Scroll down to the "*To use the cookies command*" section of this document for how to use cookies.txt in your yt-dlp command.

  Cookies.txt files will also expire eventually (sometimes after only a few minutes), and (rarely) will sometimes have their data deleted. I recommend saving cookies in a backup file to text if they're truly expired. **But I also recommend just not using this command.**

• PO Tokens (Proof of Origin)

This is another way to identify yourself to youtube's servers without looking like a bot (or at least reduce the chances). It's more complicated to acquire but still easy.

They expire in under a day, so if you ever rely on them you must get good at finding them (which is fairly easy).

I haven't

If you want to follow yt-dlp's guide, here it is. I'll try to break it down easier with screenshots, as there is technical jargon spread throughout (and this might become important in the future).

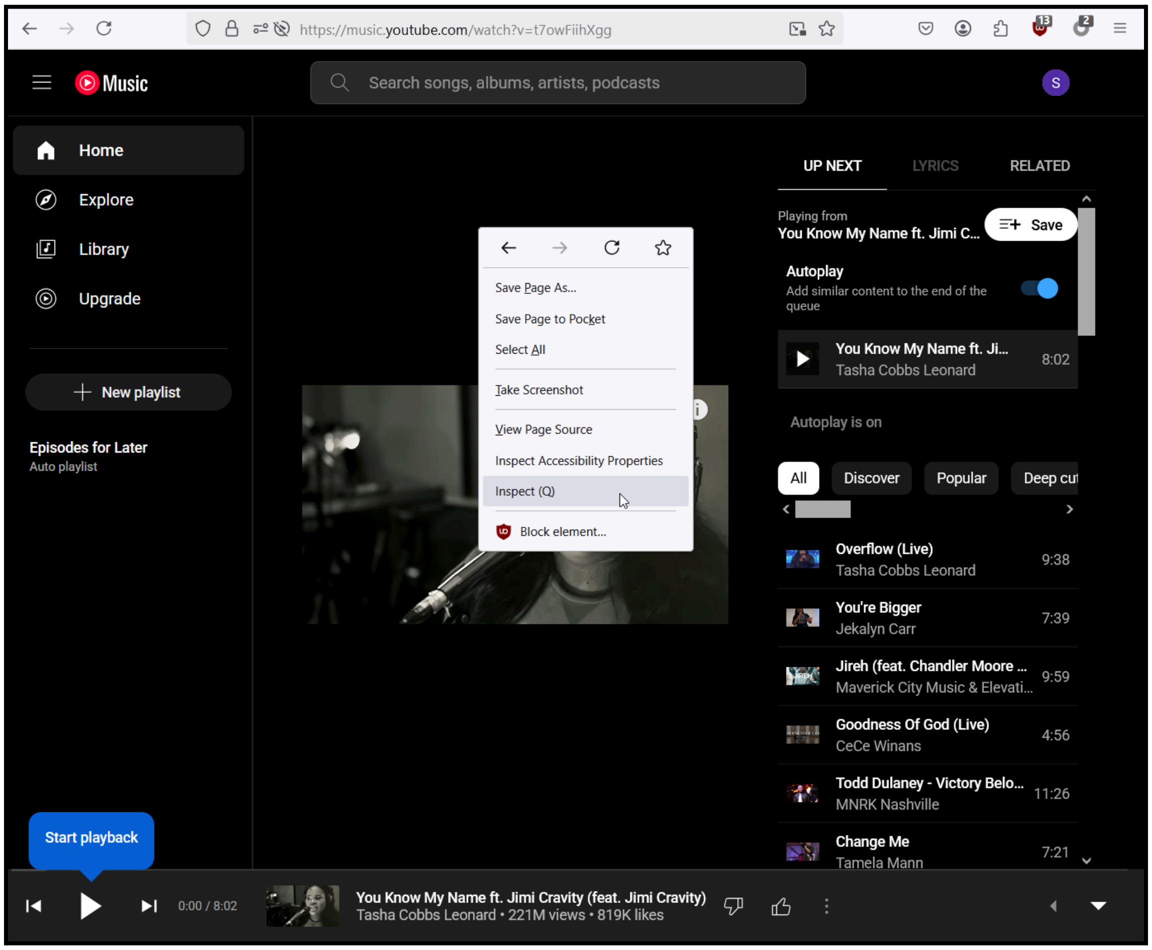
This guide will be for logged-In PO tokens (there doesn't seem to be much difference in their capabilities though), you probably want to use –cookies-from-browser with this command.

I will be using firefox in this guide but the process is almost identical in chrome/brave.

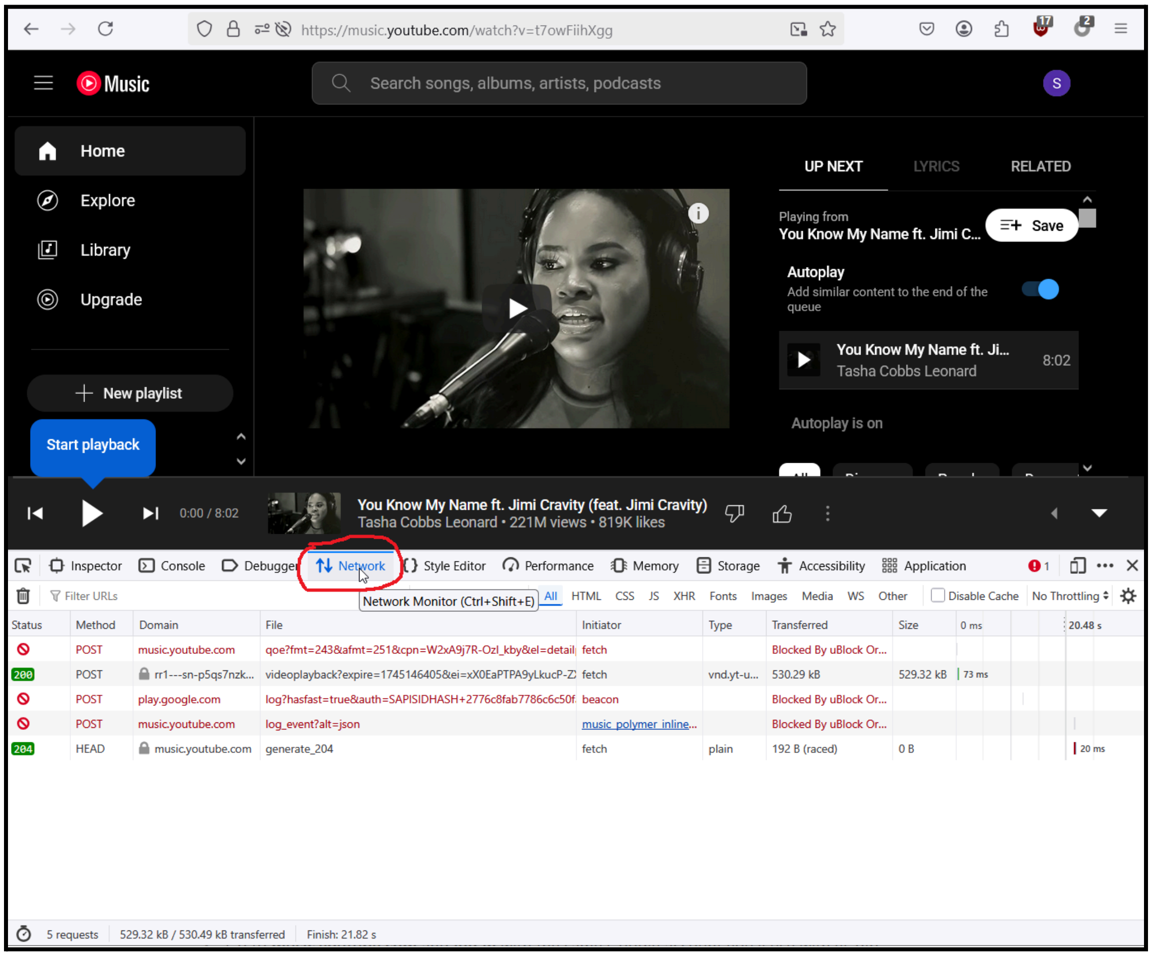
1. **Open a random Youtube Music video** in the browser you're using for the –cookies-from-browser command (logged into an account, **ideally a sock account** just in case youtube ever bans it which is unlikely).

   https://music.youtube.com/

Right click and hit "Inspect"


https://music.youtube.com/watch?v=t7owFiihXgg
Music
Search songs, albums, artists, podcasts
Home
Explore
Library
Upgrade
New playlist
Episodes for Later
Auto playlist
Save Page As...
Save Page to Pocket
Select All
Take Screenshot
View Page Source
Inspect Accessibility Properties
Inspect (Q)
Block element...
UP NEXT
LYRICS
RELATED
Playing from
You Know My Name ft. Jimi C...
Save
Autoplay
Add similar content to the end of the queue
You Know My Name ft. Ji... Tasha Cobbs Leonard 8:02
Autoplay is on
All
Discover
Popular
Deep cut
Overflow (Live) Tasha Cobbs Leonard 9:38
You're Bigger Jekalyn Carr 7:39
Jireh (feat. Chandler Moore ... Maverick City Music & Elevati... 9:59
Goodness Of God (Live) CeCe Winans 4:56
Todd Dulaney - Victory Belo... MNRK Nashville 11:26
Change Me Tamela Mann 7:21
Start playback
0:00 / 8:02
You Know My Name ft. Jimi Cravity (feat. Jimi Cravity)
Tasha Cobbs Leonard • 221M views • 819K likes

2. **Click the “Network” tab.**

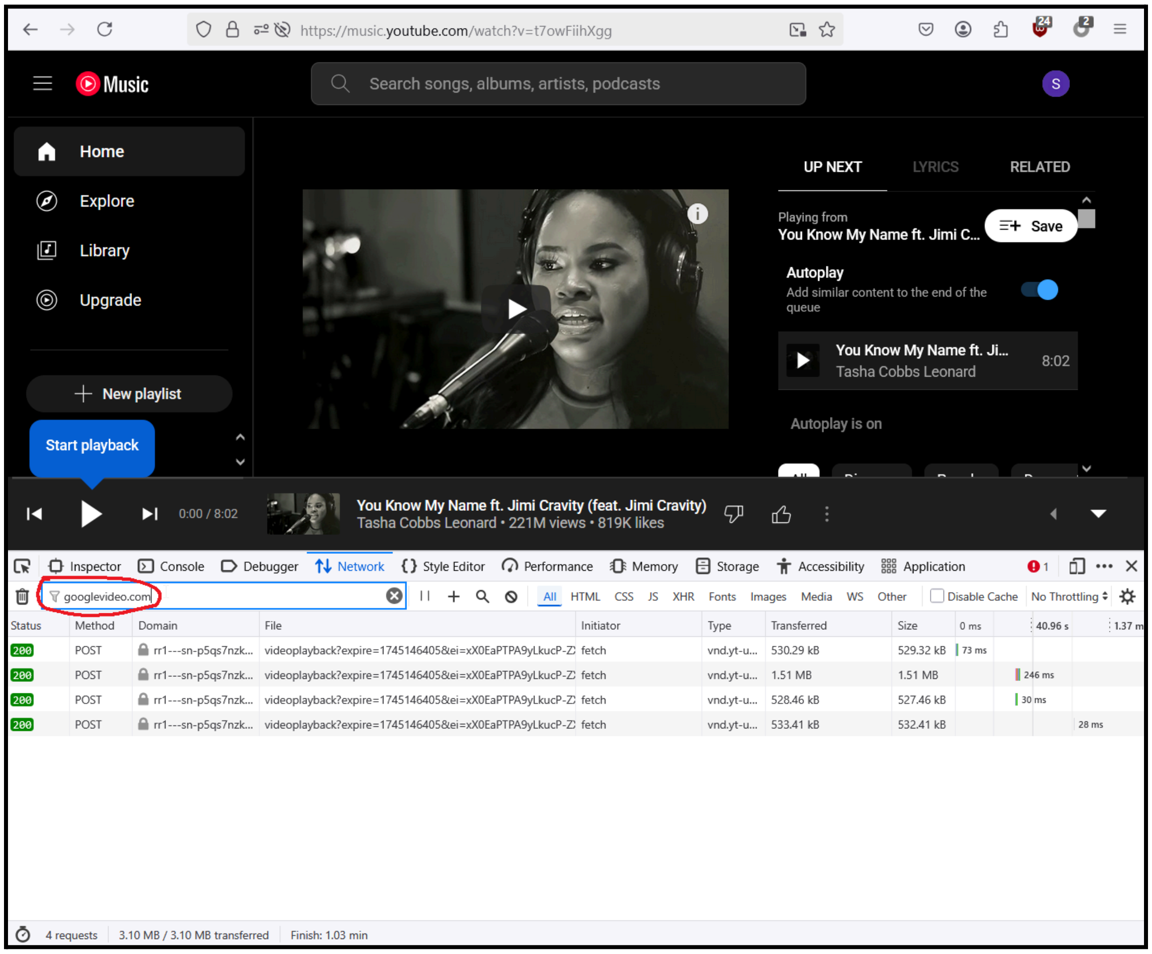


Status	Method	Domain	File	Initiator	Type	Transferred	Size	0 ms
	POST	music.youtube.com	qoe?fmt=243&afmt=251&cpn=W2xA9j7R-OzI_kby&el=detail	fetch		Blocked By uBlock Or...		
200	POST	rr1---sn-p5qs7nzk...	videoplayback?expire=1745146405&ei=xX0EaPTPA9yLkucP-Z	fetch	vnd.yt-u...	530.29 kB	529.32 kB	73 ms
	POST	play.google.com	log?hasfast=true&auth=SAPISIDHASH+2776c8fab7786c6c50f	beacon		Blocked By uBlock Or...		
	POST	music.youtube.com	log_event?alt=json	music_polymer_inline...		Blocked By uBlock Or...		
204	HEAD	music.youtube.com	generate_204	fetch	plain	192 B (raced)	0 B	20 ms

20.48 s

5 requests 529.32 kB / 530.49 kB transferred Finish: 21.82 s

3. **Click the “Filter URLs” box and type:** *googlevideo.com*



googlevideo.com

Status	Method	Domain	File	Initiator	Type	Transferred	Size	0 ms
200	POST	rr1---sn-p5qs7nzk...	videoplayback?expire=1745146405&ei=xX0EaPTPA9yLkucP-Z	fetch	vnd.yt-u...	530.29 kB	529.32 kB	73 ms
200	POST	rr1---sn-p5qs7nzk...	videoplayback?expire=1745146405&ei=xX0EaPTPA9yLkucP-Z	fetch	vnd.yt-u...	1.51 MB	1.51 MB	246 ms
200	POST	rr1---sn-p5qs7nzk...	videoplayback?expire=1745146405&ei=xX0EaPTPA9yLkucP-Z	fetch	vnd.yt-u...	528.46 kB	527.46 kB	30 ms
200	POST	rr1---sn-p5qs7nzk...	videoplayback?expire=1745146405&ei=xX0EaPTPA9yLkucP-Z	fetch	vnd.yt-u...	533.41 kB	532.41 kB	28 ms

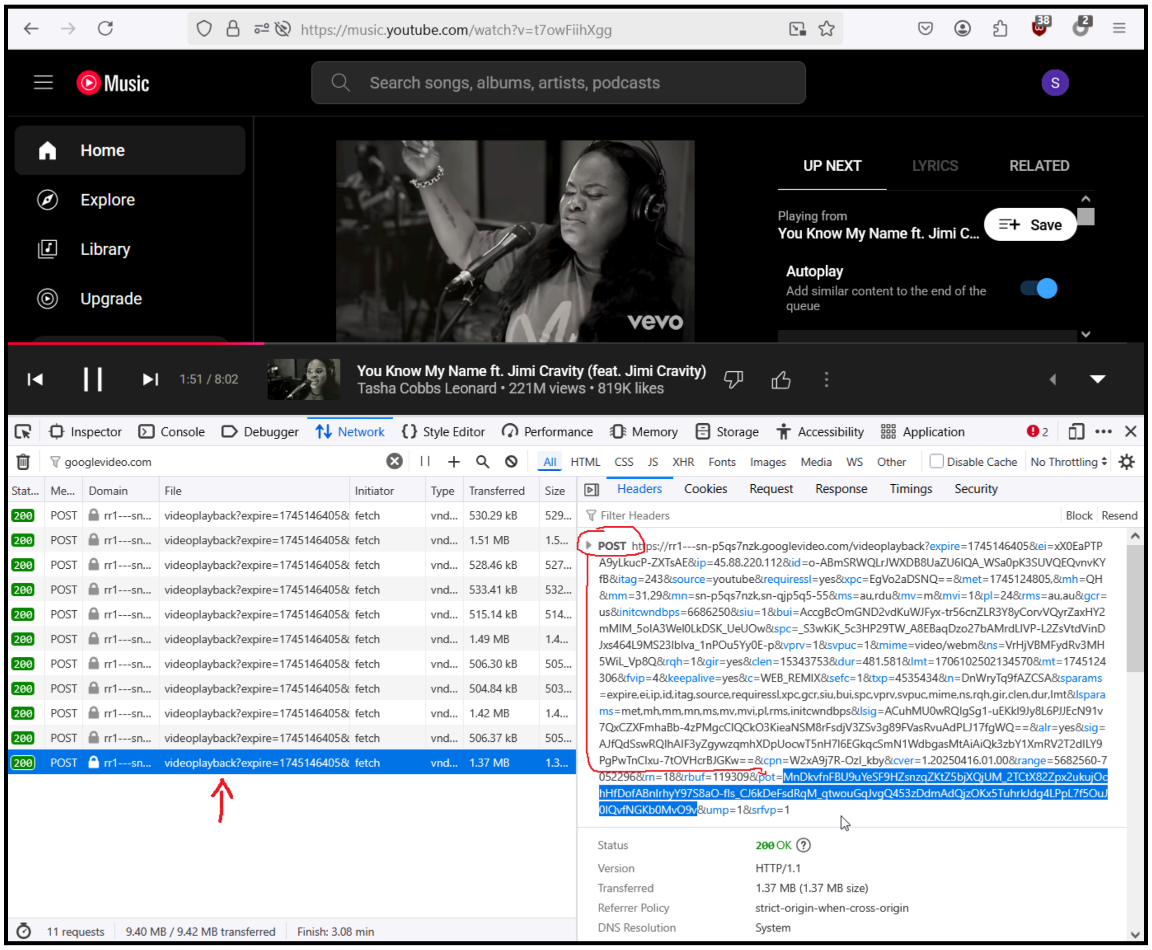
4 requests | 3.10 MB / 3.10 MB transferred | Finish: 1.03 min

4. **Play the video you have opened.**

- **Wait until a new result shows up at the bottom of the list, then click it.**

- **Find the POST section at the top, and towards the bottom of the block of text, there is a “pot=”**

- **Copy the text between “pot=” and “&”**



Stat...	Me...	Domain	File	Initiator	Type	Transferred	Size
200	POST	rr1---sn...	videoplayback?expire=1745146405&	fetch	vnd...	530.29 kB	529...
200	POST	rr1---sn...	videoplayback?expire=1745146405&	fetch	vnd...	1.51 MB	1.5...
200	POST	rr1---sn...	videoplayback?expire=1745146405&	fetch	vnd...	528.46 kB	527...
200	POST	rr1---sn...	videoplayback?expire=1745146405&	fetch	vnd...	533.41 kB	532...
200	POST	rr1---sn...	videoplayback?expire=1745146405&	fetch	vnd...	515.14 kB	514...
200	POST	rr1---sn...	videoplayback?expire=1745146405&	fetch	vnd...	1.49 MB	1.4...
200	POST	rr1---sn...	videoplayback?expire=1745146405&	fetch	vnd...	506.30 kB	505...
200	POST	rr1---sn...	videoplayback?expire=1745146405&	fetch	vnd...	504.84 kB	503...
200	POST	rr1---sn...	videoplayback?expire=1745146405&	fetch	vnd...	1.42 MB	1.4...
200	POST	rr1---sn...	videoplayback?expire=1745146405&	fetch	vnd...	506.37 kB	505...
200	POST	rr1---sn...	videoplayback?expire=1745146405&	fetch	vnd...	1.37 MB	1.3...

```
11 requests    9.40 MB / 9.42 MB transferred    Finish: 3.08 min
```

```
POST https://rr1---sn-p5qs7nzk.googlevideo.com/videoplayback?expire=1745146405&ei=xX0EaPTP
A9yLkucP-ZXTsAE&ip=45.88.220.112&id=o-ABmSRWQLrJWXDB8UaZU6IQA_WSa0pK3SUVQEQvnvKY
fB&itag=243&source=youtube&requiressl=yes&xpc=EgVo2aDSNQ==&met=1745124805,&mh=QH
&mm=31,29&mn=sn-p5qs7nzk,sn-qjp5q5-55&ms=au,rdu&mv=m&mvi=1&pl=24&rms=au,au&gcr=
us&initcwndbps=6686250&siu=1&bui=AccgBcOmGND2vdKuWJFyx-tr56cnZLR3Y8yCorvVQyrZaxHY2
mMIM_5oIA3WeI0LkDSK_UeUOw&spc=_S3wKiK_5c3HP29TW_A8EBaqDzo27bAMrdLIVP-L2ZsVtdVinD
Jxs464L9MS23Iblva_1nPOu5Yy0E-p&vprv=1&svpuc=1&mime=video/webm&ns=VrHjVBMFydRv3MH
5WiL_Vp8Q&rqh=1&gir=yes&clen=15343753&dur=481.581&lmt=1706102502134570&mt=1745124
306&fvip=4&keepalive=yes&c=WEB_REMIX&sefc=1&txp=4535434&n=DnWryTq9fAZCSA&sparams
=expire,ei,ip,id,itag,source,requiressl,xpc,gcr,siu,bui,spc,vprv,svpuc,mime,ns,rqh,gir,clen,dur,lmt&lspara
ms=met,mh,mm,mn,ms,mv,mvi,pl,rms,initcwndbps&lsig=ACuhMU0wRQIgSg1-uEKkI9Jy8L6PJJEcN91v
7QxCZXFmhaBb-4zPMgcCIQCkO3KieaNSM8rFsdjV3ZSv3g89FVasRvuAdPLJ17fgWQ==&alr=yes&sig=
AJfQdSswRQIhAIF3yZgywzqmhXDpUocwT5nH7I6EGkqcSmN1WdbgasMtAiAiQk3zbY1XmRV2T2dILY9
PgPwTnClxu-7tOVHcrBJGKw==&cpn=W2xA9j7R-OzI_kby&cver=1.20250416.01.00&range=5682560-7
052296&rn=18&rbuf=119309&pot=MnDkvfnFBU9uYeSF9HZsnzqZKtZ5bjXQjUM_2TCtX82Zpx2ukujOc
hHfDofABnIrhyY97S8aO-fls_CJ6kDeFsdRqM_qtwouGqJvgQ453zDdmAdQjzOKx5TuhrkJdg4LPpL7f5OuJ
0IQvfNGKb0MvO9v&ump=1&srfvp=1
```

```
Status             200 OK
Version            HTTP/1.1
Transferred        1.37 MB (1.37 MB size)
Referrer Policy    strict-origin-when-cross-origin
DNS Resolution     System
```

5. **The text you copied can be inserted into this yt-dlp command:**

*--extractor-args "youtube:po_token=web.gvs+**YOUR_GVS_TOKEN**"*

Here is a fake PO token for example.

--extractor-args "youtube:po_token=web.gvs+MnDkvfnFBU9uYeSF9HZsnzqZKtZ5bjXQjUM_2TCtX82Zpx6ukujFchHfDgfABnIrhyY97V8aO-fls_CJ6kDeFsdRqM_qtwouGqJvgQ453zDdmAdQjzOKx5TuhrkJdg4LPpL7f5OuJ0IQvfNGKb0MvO9v"

6.

DRM Format Warning

This is a new error as of March-April 2025, where yt-dlp downloads will show this error and proceed to download a 360p quality because the TV client formats have been temporarily blocked.

If you see this warning, a single time and it *continues* to download the video, you need to change your default command to have this:

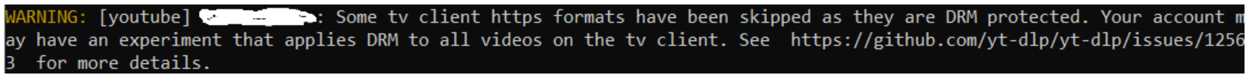
`-f "bv+ba"`

Adding this argument will let the warning count as an error, which is important.

If you let it continue downloading, your video(s) will be in EXTREMELY low quality.

And it will count as downloaded too (with the –download-archive command), making it difficult to re-download at the correct quality without re-making the download archive file, or removing the video IDs from the file).

This is mainly happening on attempts at downloading large youtube channels / multiple videos in a row (on top of the “Sign In” errors, and “Video Unavailable” errors which happen after a certain amount of time of downloading in a single command).



```
WARNING: [youtube] [redacted]: Some tv client https formats have been skipped as they are DRM protected. Your account m
ay have an experiment that applies DRM to all videos on the tv client. See  https://github.com/yt-dlp/yt-dlp/issues/1256
3  for more details.
```

If you see this error, STOP THE COMMAND IMMEDIATELY. You will end up downloading 360p quality, and if you are running the –download-archive command the video will save to the archive file and won't redownload.

This will toxify your attempt at using the program (if it happens).

If you suspect multiple of your files ended up at 360p, you will have to use a new script to detect these files by resolution, delete them, and automatically remove those IDs from the download archive file.
(note: improve this)

I have noticed this is not IP based either, I can only fix it by either waiting a certain amount of time (there is a timeout) or using a new browser profile. **My browser profiles are getting "DRM Blacklisted" after about 200 video downloads for example, and those profiles won't re-download. They also get Sign In errors, but tab refreshers fix that.**

Cookie Auto-Rotation Script (Solves all errors)

Here is a python script which runs yt-dlp within it, automatically detects the DRM warnings and terminates the command, and rotates between different browser profile locations.

*The DRM warning skip thing isn't relevant now that I found this command **-f "bv+ba"**, but you can still take advantage of the rotating browser profile cookies **ESPECIALLY** if you're mass-downloading.*

The fact the script can use multiple different alternating commands lets you, effectively, never get blocked.

This is the only way I can think of to mass-archive a large channel anymore, because the blacklisting has gotten so strong, you can regularly run into 3 different errors.

You probably want to keep cycling between 5 or so browser profiles, once all five of them get blacklisted, delete those 5, create another 5 all at once (with the same tab refresher, log into youtube on all 5 of them). Rinse and repeat.

Download (single URL): Auto_yt-dlp_cookies_rotation_mkii.py

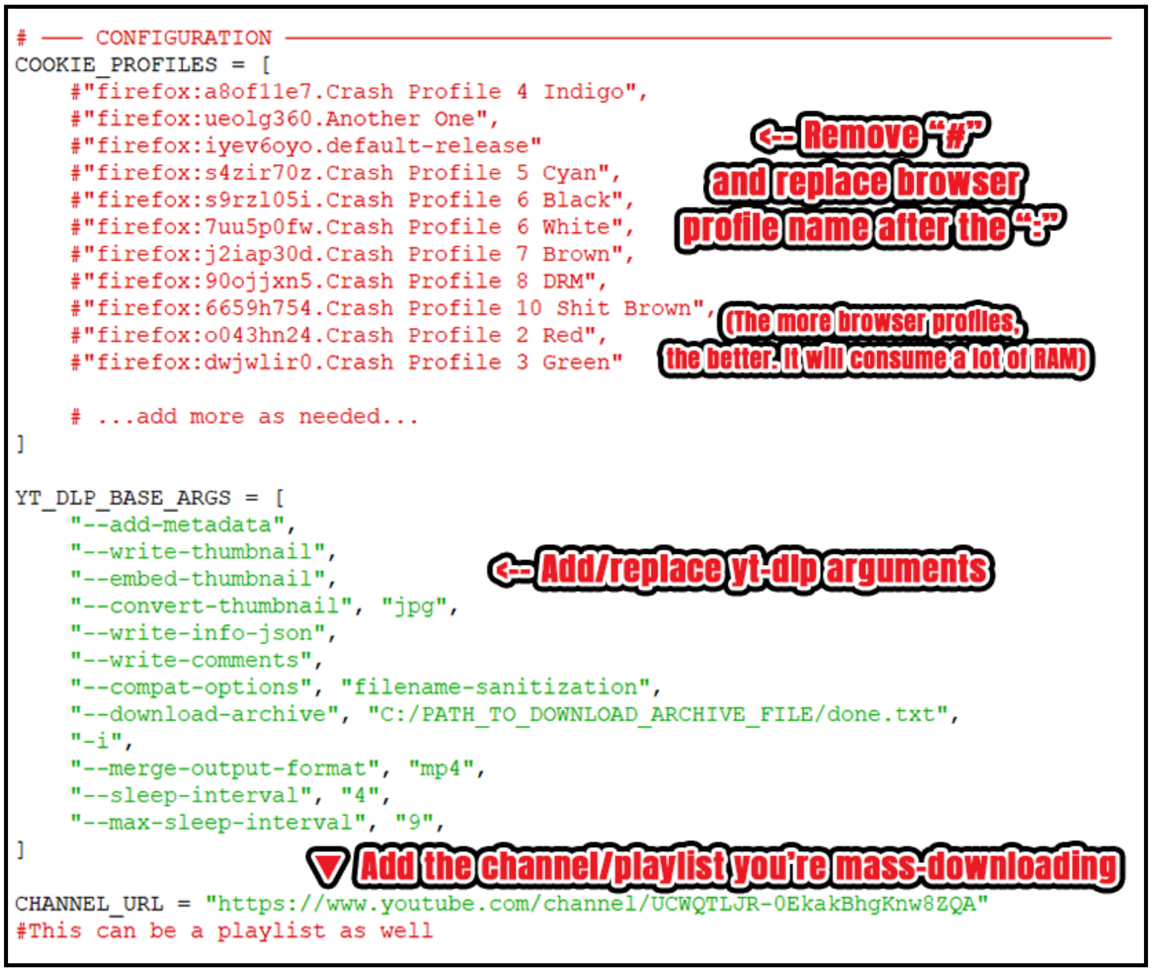
Download (multiple URLs, useful for auto-archiving many channels in 1 folder):

Auto_yt-dlp_cookies_mkiv_multiurls.py

^^ Theoretically, you could configure this script to download certain channel URLs within certain sub-folders (based on channel ID). This is possible with AI coding, I may do it eventually. I do have a script which can auto-organize files by channel name though

1. **Refer to the "SPECIAL TOOLS" header to run this python script using Python IDLE, this is very easy to set up, and unfortunately necessary.**

2. **Add in your firefox browser profiles, you created from the previous section, into the python script.**



<-- Remove "#"
and replace browser
profile name after the ":"
(The more browser profiles,
the better. It will consume a lot of RAM)
<-- Add/replace yt-dlp arguments
Add the channel/playlist you're mass-downloading

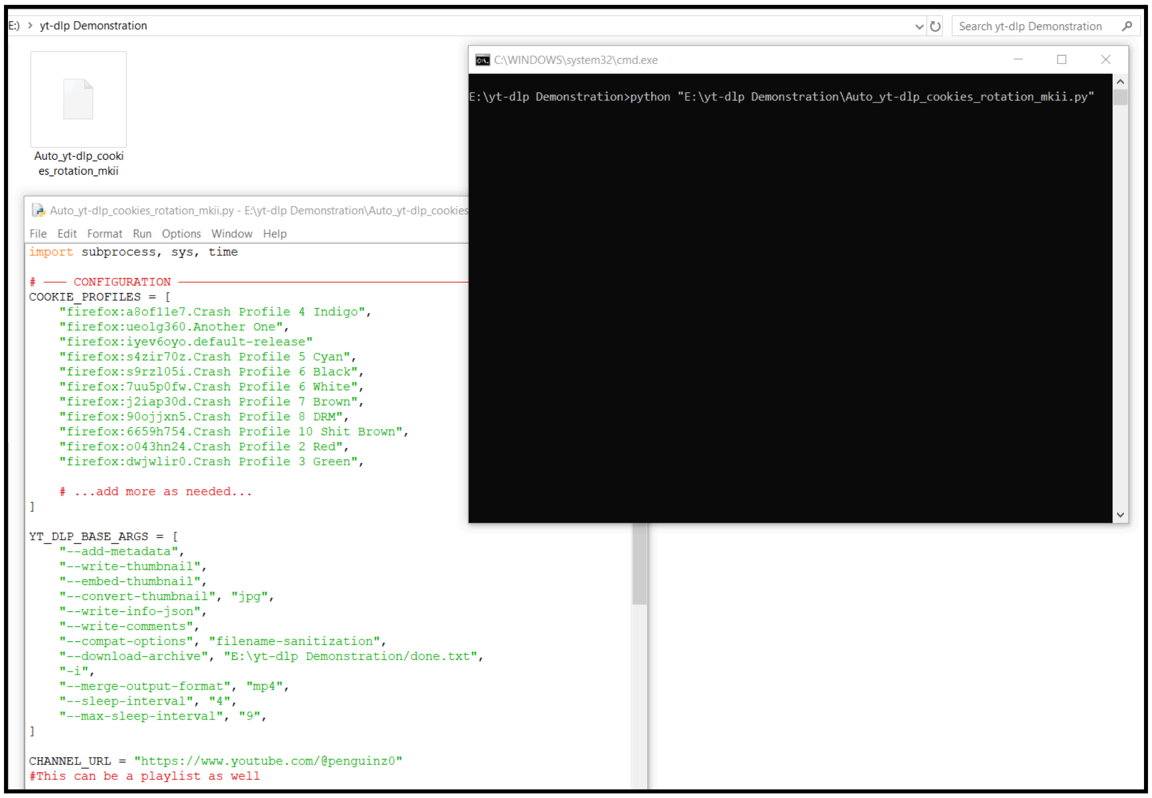
```
# ---- CONFIGURATION -------------------------------------------------------
COOKIE_PROFILES = [
    #"firefox:a8of11e7.Crash Profile 4 Indigo",
    #"firefox:ueolg360.Another One",
    #"firefox:iyev6oyo.default-release"
    #"firefox:s4zir70z.Crash Profile 5 Cyan",
    #"firefox:s9rzl05i.Crash Profile 6 Black",
    #"firefox:7uu5p0fw.Crash Profile 6 White",
    #"firefox:j2iap30d.Crash Profile 7 Brown",
    #"firefox:90ojjxn5.Crash Profile 8 DRM",
    #"firefox:6659h754.Crash Profile 10 Shit Brown",
    #"firefox:o043hn24.Crash Profile 2 Red",
    #"firefox:dwjwlir0.Crash Profile 3 Green"

    # ...add more as needed...
]

YT_DLP_BASE_ARGS = [
    "--add-metadata",
    "--write-thumbnail",
    "--embed-thumbnail",
    "--convert-thumbnail", "jpg",
    "--write-info-json",
    "--write-comments",
    "--compat-options", "filename-sanitization",
    "--download-archive", "C:/PATH_TO_DOWNLOAD_ARCHIVE_FILE/done.txt",
    "-i",
    "--merge-output-format", "mp4",
    "--sleep-interval", "4",
    "--max-sleep-interval", "9",
]

CHANNEL_URL = "https://www.youtube.com/channel/UCWQTLJR-0EkakBhgKnw8ZQA"
#This can be a playlist as well
```

3. **Add in/replace your yt-dlp arguments, and then the playlist/channel URL you're trying to download.**

   "–download-archive" is MANDATORY*, or this script will keep re-downloading videos.*

4. **Call the script from within the directory you want to run the script.**

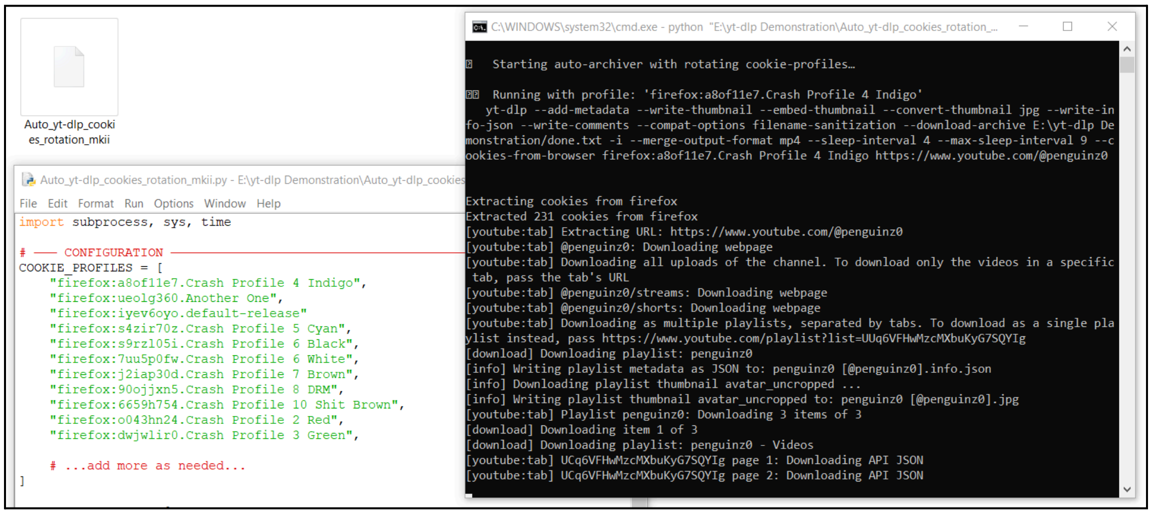
   *python "**E:\yt-dlp Demonstration**\Auto_yt-dlp_cookies_rotation_mkii.py"*


E:) > yt-dlp Demonstration
Search yt-dlp Demonstration
Auto_yt-dlp_cookies_rotation_mkii
Auto_yt-dlp_cookies_rotation_mkii.py - E:\yt-dlp Demonstration\Auto_yt-dlp_cookies
File Edit Format Run Options Window Help
C:\WINDOWS\system32\cmd.exe

```
E:\yt-dlp Demonstration>python "E:\yt-dlp Demonstration\Auto_yt-dlp_cookies_rotation_mkii.py"
```

```python
import subprocess, sys, time

# ── CONFIGURATION ──────────────────────────
COOKIE_PROFILES = [
    "firefox:a8of11e7.Crash Profile 4 Indigo",
    "firefox:ueolg360.Another One",
    "firefox:iyev6oyo.default-release"
    "firefox:s4zir70z.Crash Profile 5 Cyan",
    "firefox:s9rzl05i.Crash Profile 6 Black",
    "firefox:7uu5p0fw.Crash Profile 6 White",
    "firefox:j2iap30d.Crash Profile 7 Brown",
    "firefox:90ojjxn5.Crash Profile 8 DRM",
    "firefox:6659h754.Crash Profile 10 Shit Brown",
    "firefox:o043hn24.Crash Profile 2 Red",
    "firefox:dwjwlir0.Crash Profile 3 Green",

    # ...add more as needed...
]

YT_DLP_BASE_ARGS = [
    "--add-metadata",
    "--write-thumbnail",
    "--embed-thumbnail",
    "--convert-thumbnail", "jpg",
    "--write-info-json",
    "--write-comments",
    "--compat-options", "filename-sanitization",
    "--download-archive", "E:\yt-dlp Demonstration/done.txt",
    "-i",
    "--merge-output-format", "mp4",
    "--sleep-interval", "4",
    "--max-sleep-interval", "9",
]

CHANNEL_URL = "https://www.youtube.com/@penguinz0"
#This can be a playlist as well
```

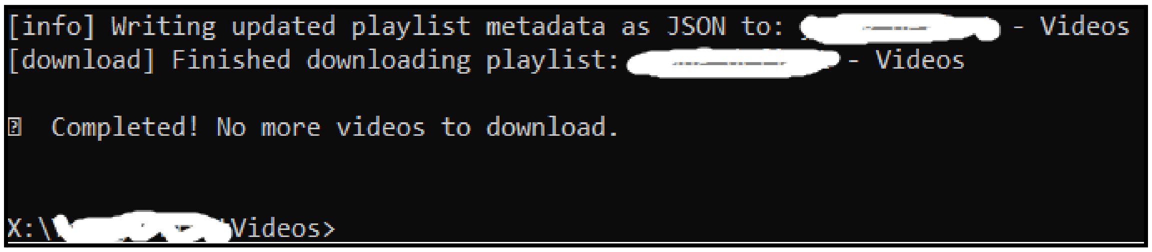
If it works correctly, you will see “Starting auto-archiver” and then “Running with profile:”


Auto_yt-dlp_cookies_rotation_mkii
Auto_yt-dlp_cookies_rotation_mkii.py - E:\yt-dlp Demonstration\Auto_yt-dlp_cookies
File Edit Format Run Options Window Help
C:\WINDOWS\system32\cmd.exe - python "E:\yt-dlp Demonstration\Auto_yt-dlp_cookies_rotation_...

```python
import subprocess, sys, time

# ── CONFIGURATION ──────────────────────────
COOKIE_PROFILES = [
    "firefox:a8of11e7.Crash Profile 4 Indigo",
    "firefox:ueolg360.Another One",
    "firefox:iyev6oyo.default-release"
    "firefox:s4zir70z.Crash Profile 5 Cyan",
    "firefox:s9rzl05i.Crash Profile 6 Black",
    "firefox:7uu5p0fw.Crash Profile 6 White",
    "firefox:j2iap30d.Crash Profile 7 Brown",
    "firefox:90ojjxn5.Crash Profile 8 DRM",
    "firefox:6659h754.Crash Profile 10 Shit Brown",
    "firefox:o043hn24.Crash Profile 2 Red",
    "firefox:dwjwlir0.Crash Profile 3 Green",

    # ...add more as needed...
]
```

```
?   Starting auto-archiver with rotating cookie-profiles…

??  Running with profile: 'firefox:a8of11e7.Crash Profile 4 Indigo'
  yt-dlp --add-metadata --write-thumbnail --embed-thumbnail --convert-thumbnail jpg --write-in
fo-json --write-comments --compat-options filename-sanitization --download-archive E:\yt-dlp De
monstration/done.txt -i --merge-output-format mp4 --sleep-interval 4 --max-sleep-interval 9 --c
ookies-from-browser firefox:a8of11e7.Crash Profile 4 Indigo https://www.youtube.com/@penguinz0


Extracting cookies from firefox
Extracted 231 cookies from firefox
[youtube:tab] Extracting URL: https://www.youtube.com/@penguinz0
[youtube:tab] @penguinz0: Downloading webpage
[youtube:tab] Downloading all uploads of the channel. To download only the videos in a specific
 tab, pass the tab's URL
[youtube:tab] @penguinz0/streams: Downloading webpage
[youtube:tab] @penguinz0/shorts: Downloading webpage
[youtube:tab] Downloading as multiple playlists, separated by tabs. To download as a single pla
ylist instead, pass https://www.youtube.com/playlist?list=UUq6VFHwMzcMXbuKyG7SQYIg
[download] Downloading playlist: penguinz0
[info] Writing playlist metadata as JSON to: penguinz0 [@penguinz0].info.json
[info] Downloading playlist thumbnail avatar_uncropped ...
[info] Writing playlist thumbnail avatar_uncropped to: penguinz0 [@penguinz0].jpg
[youtube:tab] Playlist penguinz0: Downloading 3 items of 3
[download] Downloading item 1 of 3
[download] Downloading playlist: penguinz0 - Videos
[youtube:tab] UCq6VFHwMzcMXbuKyG7SQYIg page 1: Downloading API JSON
[youtube:tab] UCq6VFHwMzcMXbuKyG7SQYIg page 2: Downloading API JSON
```

5. This script will loop infinitely, alternating between different browser profiles as soon as a DRM, Sign In, or Video Unavailable error is detected, until all of the videos are downloaded.



```
[info] Writing updated playlist metadata as JSON to: [redacted] - Videos
[download] Finished downloading playlist: [redacted] - Videos

?  Completed! No more videos to download.


X:\[redacted]Videos>
```

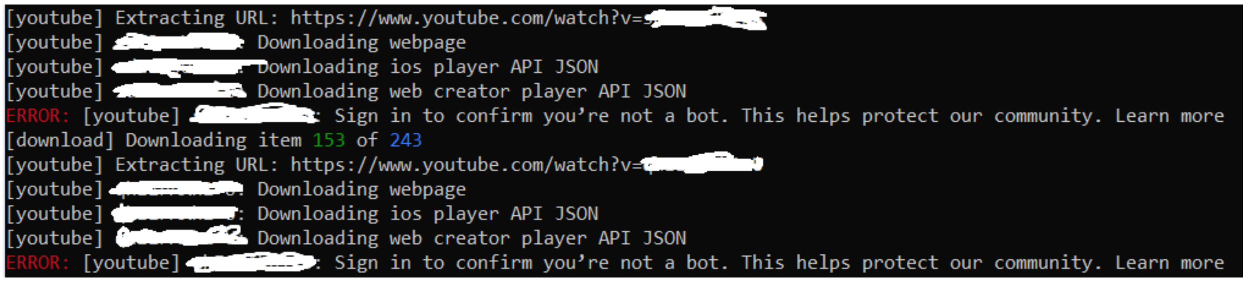
Currently the script doesn't skip other cases like member's only videos, leading the script to loop infinitely. This will be fixed soon.

List of yt-dlp Blacklist Errors

Ever since 2024, using yt-dlp to download more than a few dozen files has become an increasingly complicated and infuriating task.

Your results may vary, but at the least you should be aware of them and know where to look to fix them.

"Sign In" Error



```
[youtube] Extracting URL: https://www.youtube.com/watch?v=[redacted]
[youtube] [redacted] Downloading webpage
[youtube] [redacted]Downloading ios player API JSON
[youtube] [redacted] Downloading web creator player API JSON
ERROR: [youtube] [redacted]: Sign in to confirm you’re not a bot. This helps protect our community. Learn more
[download] Downloading item 153 of 243
[youtube] Extracting URL: https://www.youtube.com/watch?v=[redacted]
[youtube] [redacted]: Downloading webpage
[youtube] [redacted]: Downloading ios player API JSON
[youtube] [redacted] Downloading web creator player API JSON
ERROR: [youtube][redacted]: Sign in to confirm you’re not a bot. This helps protect our community. Learn more
```

This is the most basic and common error you will see using yt-dlp. Since around June 2024, Youtube can now temporarily blacklist your IP, and the cookies you're using to download videos (cookies may just expire, I’m not sure).

If you use a VPN, you are <u>effectively guaranteed</u> to see this error at some point (if not immediately, after downloading several videos), and must address it unless you’re downloading only a few videos at a time. Sometimes, your VPN IP is randomly blacklisted anyway without downloading anything, and you’ll need to use cookies.

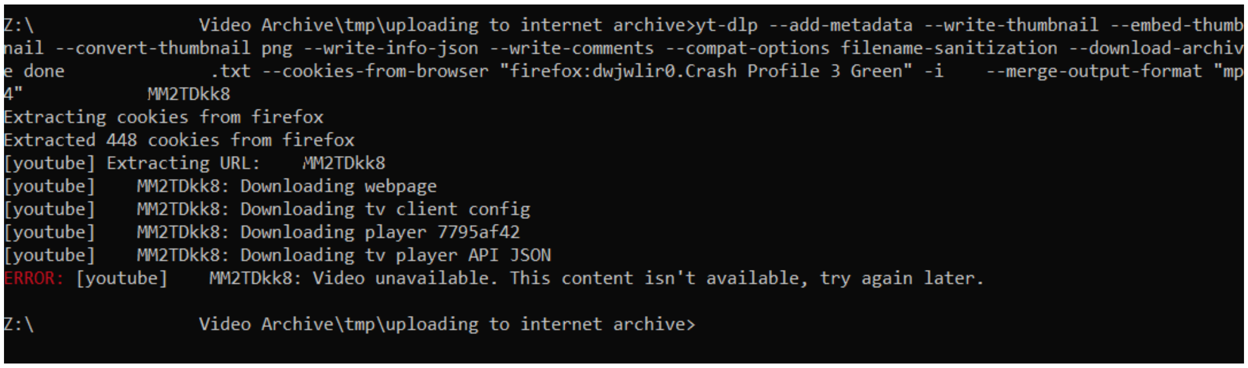
If you use your residential IP, you may only see this error after a long downloading session, but you will see it at some point.

<u>There are several reported solutions, but this is the ideal one for me:</u>

- **Use –cookies-from-browser “firefox:BROWSERPROFILENAME”. You must have that firefox profile open, with a signed-in youtube window which you have a tab refresher extension (scroll up).**

You will still see this error once your cookies expire / or are blacklisted, but as long as they’re refreshing automatically, this issue won’t hold you back. The next one will.

“Video Unavailable” Error



```
Z:\               Video Archive\tmp\uploading to internet archive>yt-dlp --add-metadata --write-thumbnail --embed-thumb
nail --convert-thumbnail png --write-info-json --write-comments --compat-options filename-sanitization --download-archiv
e done             .txt --cookies-from-browser "firefox:dwjwlir0.Crash Profile 3 Green" -i    --merge-output-format "mp
4"          MM2TDkk8
Extracting cookies from firefox
Extracted 448 cookies from firefox
[youtube] Extracting URL:    MM2TDkk8
[youtube]    MM2TDkk8: Downloading webpage
[youtube]    MM2TDkk8: Downloading tv client config
[youtube]    MM2TDkk8: Downloading player 7795af42
[youtube]    MM2TDkk8: Downloading tv player API JSON
ERROR: [youtube]    MM2TDkk8: Video unavailable. This content isn't available, try again later.

Z:\               Video Archive\tmp\uploading to internet archive>
```

Consider this a more severe blacklist. I have noticed this error after long downloading sessions using the same –cookies-from-browser command.

Importantly, this will happen REGARDLESS of if you’re running a tab refresher.

There are only two main solutions I’ve seen:

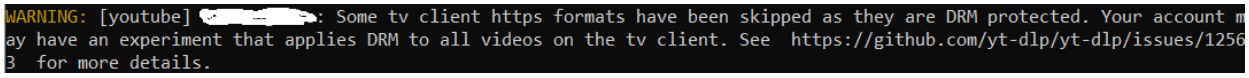
- Switch to a new browser profile to use –browser-cookies from. This will always fix the issue.

  Ideally, you can use the Cookie Auto-Rotation script (scroll up) with multiple firefox profiles open at once, so it can continually run your command until it downloads all the videos. (If you notice your firefox profile keeps getting a DRM protected error, you might have to stop using it and replace it with a new one).

- **Wait a few hours or a day or two until the blacklist expires (it always does, to my knowledge).**

“DRM Protected” Warning

(New as of March 2025)



```
WARNING: [youtube] [redacted]: Some tv client https formats have been skipped as they are DRM protected. Your account m
ay have an experiment that applies DRM to all videos on the tv client. See  https://github.com/yt-dlp/yt-dlp/issues/1256
3  for more details.
```

This is another more severe blacklist, and it’s happening more regularly now. For me, it's happening regularly starting April 17th 2025.

While this only acts like a warning, you NEED to consider it an error and immediately terminate the program when you see it, otherwise you will download files at 360p. This is why I made a script which can do this automatically.

While I don’t know if it’s VPN/proxy related, I do know that the list of solutions is even lower.

There are only two main solutions I’ve seen:

DISCLAIMER FOR VIDEO EDITORS

yt-dlp does not download in a video editor compatible format by default; neither do many video downloaders. You could blame this on Adobe and Sony Vegas for not supporting modern codec and audio formats, like websites not supporting .webp image formats.

But mainly blame google for creating formats nobody uses until they force them upon you. They're slowly phasing out h264 formats on all Youtube videos, which has been reported to possibly destroy the video quality of old videos.

If you use DaVinci Resolve or CapCut, this doesn't apply to you. If you're a linux user it probably won't apply to you. If you're doing basic editing, it might be ideal to just use that with yt-dlp videos, rather than continue reading.

yt-dlp will default to download with the VP9 video codec and OPUS audio. This will not work in most video editors by default, specifically the *OPUS audio format*. There is (*almost*) no way to download in a different format, plain and simple. Unless you want 360p resolution. Maybe there's a way to change the embedded audio format but I'm unaware, vp9 would still be a major problem even if OPUS audio was fixed.

There are ways to specify certain formats for videos and possibly get h264+AAC, but this is not a universal application like *not specifying* the format usually is (to my knowledge, could be wrong).

Also, if for whatever that format is not available on the video, you will get an error, so it may not be a universally applicable command.

The MIRACLE Solution (Risky?)

Warning: *This involves downloading a crack which (according to Windows Defender and several other anti-viruses) is labelled as malware. MalwareBytes also detects various registry edits after installing. So this may be retarded, but worth the risk to some people, if you are willing to trust my intentions.*

It is fairly common for real cracks to be false-flagged as malware, as a result of how genuine cracks behave and/or for anti-viruses to scare people off from using them, and this is known in piracy communities. It's also possible the registry edits, which are flagged, are a side effect of how the crack works.

I am aware how reckless this seems to place here, but all I can say is it was out of desperation, and because this is the only solution which worked perfectly for me. I've used it for a few months now and it's worked fine*. I want to be as transparent about it as possible. I'm trying to share the best solutions to my problems.*

So far, and it's been about 5 months, I've had no PC issues like further malware detections on my PC. For me, it's fairly clear it's safe. For you, you might have to weigh the risks.

Or use Davinci Resolve / Capcut, or re-encode every yt-dlp video.

For Adobe Premiere users who want to use yt-dlp, this is the one and only solution I was able to find. And luckily it **works perfectly**. Minus the warnings and flags for malware, you'd have to overlook that.

There happens to be a 3rd-party plugin (Influx) which allows Adobe Premiere to work with VP9, MKV, and OPUS audio for which Adobe is too lazy and greedy to implement like a worthwhile company would.

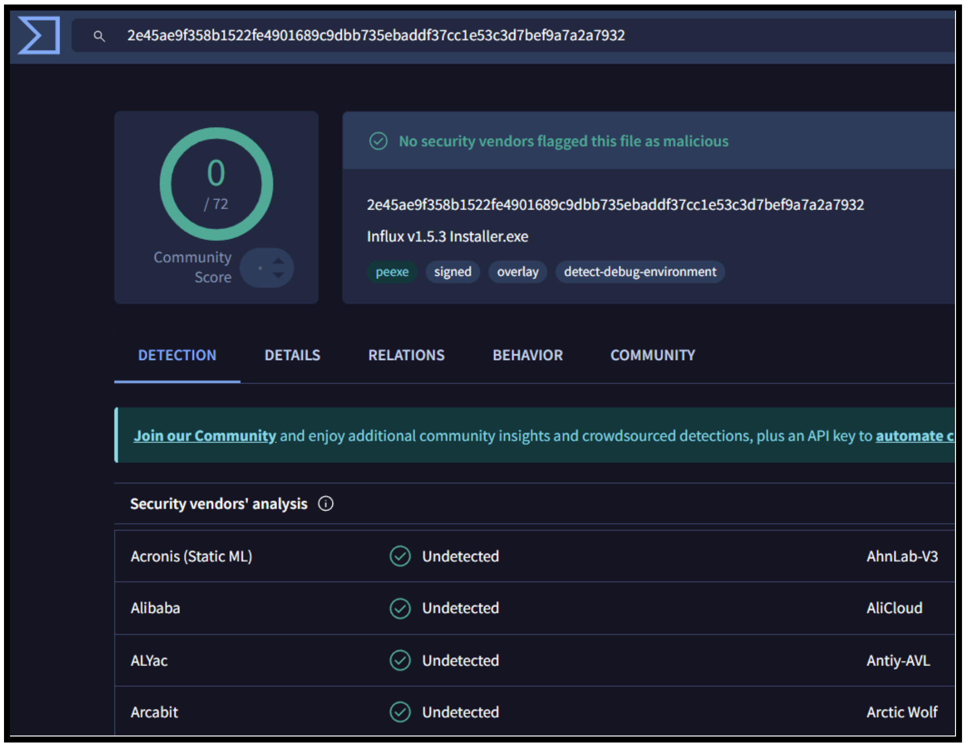
This plugin is a flat $90 fee, for something no one should have to pay to deal with. Luckily a crack exists, but it's a hard sell.

I would advise you get a m0nkrus crack for Adobe Premiere 2025 as well, the website is well regarded on large piracy subreddits like Pgen (for Adobe Creative Cloud) which has over 200K members (edit: it just got banned, funnily enough, what pieces of shit). I have installed Adobe 2023, Photoshop, and Adobe 2025 and it just works instantly (minus the occasional adobe warning which you'd have to patch.)

Influx 1.5.3 for Adobe 2025 (Install at your own risk)

To verify a crack is safe you can usually run the file through VirusTotal for hidden malware unless it is over a few gigabytes.

The .exe file which you run to install Influx is safe, as expected since it's the official installer.


2e45ae9f358b1522fe4901689c9dbb735ebaddf37cc1e53c3d7bef9a7a2a7932

0 / 72

Community Score

No security vendors flagged this file as malicious

2e45ae9f358b1522fe4901689c9dbb735ebaddf37cc1e53c3d7bef9a7a2a7932

Influx v1.5.3 Installer.exe

peexe signed overlay detect-debug-environment

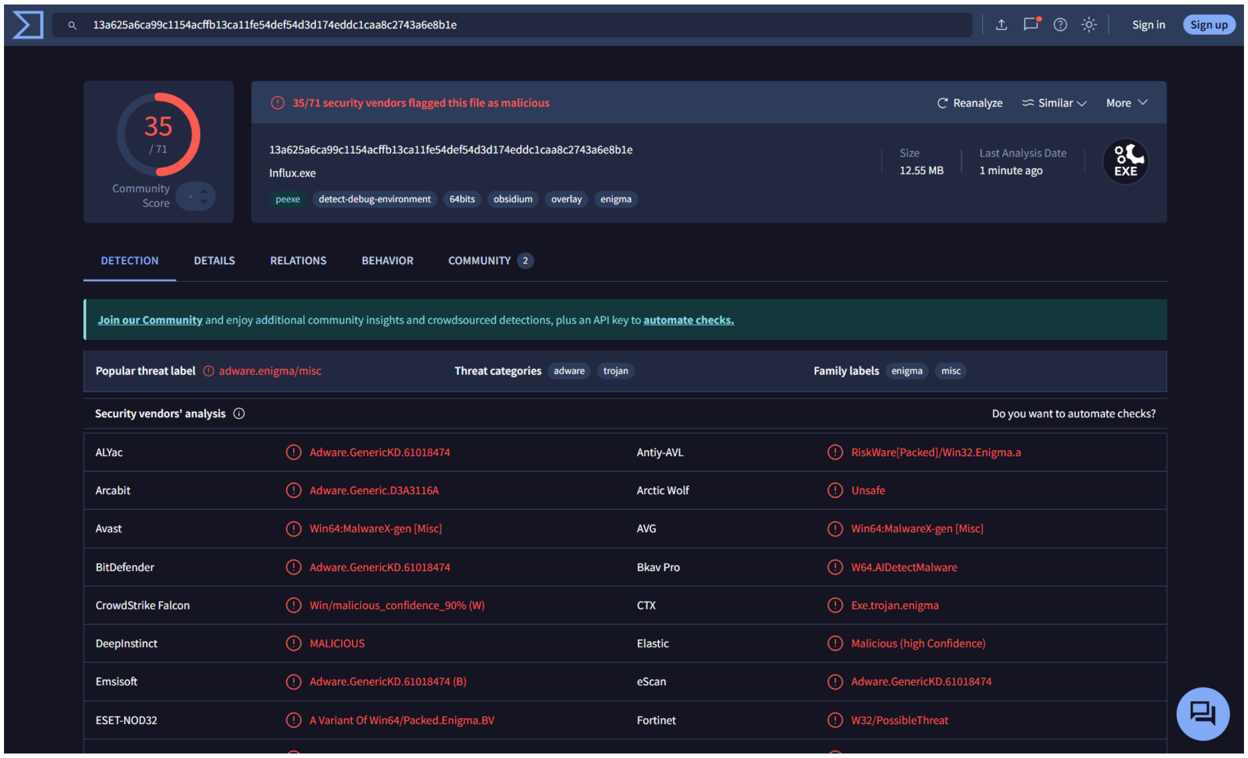
DETECTION DETAILS RELATIONS BEHAVIOR COMMUNITY

Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate c

Security vendors' analysis

Acronis (Static ML)	Undetected	AhnLab-V3
Alibaba	Undetected	AliCloud
ALYac	Undetected	Antiy-AVL
Arcabit	Undetected	Arctic Wolf

There is a file “influx.exe” located in the “Autokroma Influx” directory. This file not only is detected in half of VirusTotal’s vendors, it also will be detected by Windows Defender when run on the directory.



13a625a6ca99c1154acffb13ca11fe54def54d3d174eddc1caa8c2743a6e8b1e

35 / 71 Community Score

35/71 security vendors flagged this file as malicious

13a625a6ca99c1154acffb13ca11fe54def54d3d174eddc1caa8c2743a6e8b1e
Influx.exe
peexe detect-debug-environment 64bits obsidium overlay enigma

Size 12.55 MB
Last Analysis Date 1 minute ago

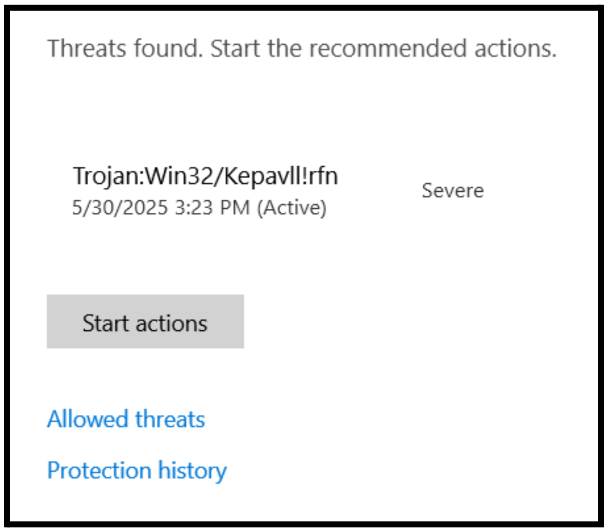
DETECTION DETAILS RELATIONS BEHAVIOR COMMUNITY 2

Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label adware.enigma/misc — Threat categories adware trojan — Family labels enigma misc

Security vendors' analysis — Do you want to automate checks?

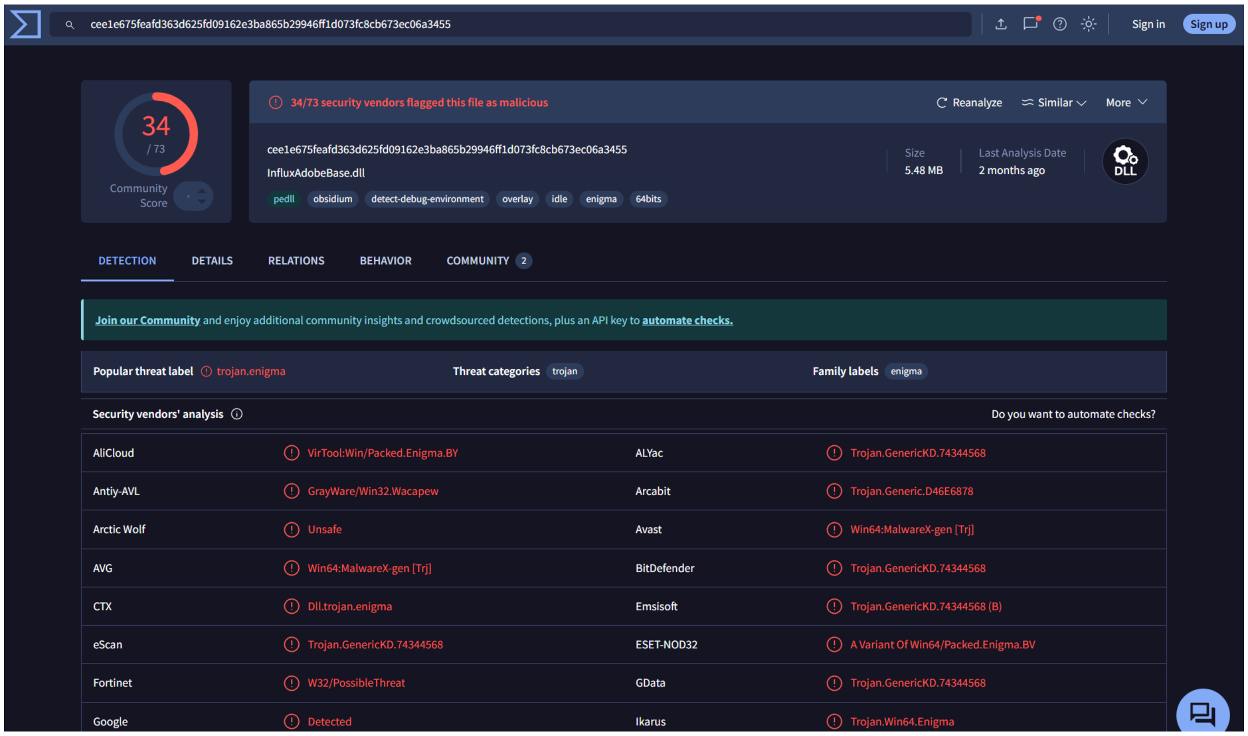
ALYac	Adware.GenericKD.61018474	Antiy-AVL	RiskWare[Packed]/Win32.Enigma.a
Arcabit	Adware.Generic.D3A3116A	Arctic Wolf	Unsafe
Avast	Win64:MalwareX-gen [Misc]	AVG	Win64:MalwareX-gen [Misc]
BitDefender	Adware.GenericKD.61018474	Bkav Pro	W64.AIDetectMalware
CrowdStrike Falcon	Win/malicious_confidence_90% (W)	CTX	Exe.trojan.enigma
DeepInstinct	MALICIOUS	Elastic	Malicious (high Confidence)
Emsisoft	Adware.GenericKD.61018474 (B)	eScan	Adware.GenericKD.61018474
ESET-NOD32	A Variant Of Win64/Packed.Enigma.BV	Fortinet	W32/PossibleThreat



Threats found. Start the recommended actions.

Trojan:Win32/Kepavll!rfn
5/30/2025 3:23 PM (Active) — Severe

On top of that, **a required file to run the plugin “InfluxAdobeBase.dll” gives the same detections.**



34/73 security vendors flagged this file as malicious

cee1e675feafd363d625fd09162e3ba865b29946ff1d073fc8cb673ec06a3455
InfluxAdobeBase.dll
pedll obsidium detect-debug-environment overlay idle enigma 64bits

Size
5.48 MB

Last Analysis Date
2 months ago

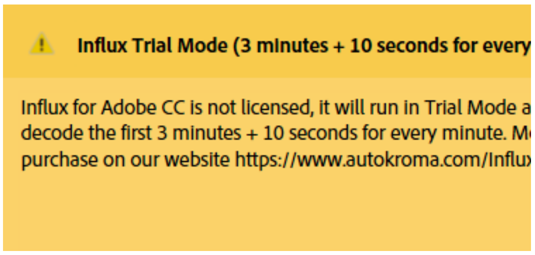
Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label trojan.enigma   Threat categories trojan   Family labels enigma

Security vendors' analysis   Do you want to automate checks?

AliCloud	VirTool:Win/Packed.Enigma.BY	ALYac	Trojan.GenericKD.74344568
Antiy-AVL	GrayWare/Win32.Wacapew	Arcabit	Trojan.Generic.D46E6878
Arctic Wolf	Unsafe	Avast	Win64:MalwareX-gen [Trj]
AVG	Win64:MalwareX-gen [Trj]	BitDefender	Trojan.GenericKD.74344568
CTX	Dll.trojan.enigma	Emsisoft	Trojan.GenericKD.74344568 (B)
eScan	Trojan.GenericKD.74344568	ESET-NOD32	A Variant Of Win64/Packed.Enigma.BV
Fortinet	W32/PossibleThreat	GData	Trojan.GenericKD.74344568
Google	Detected	Ikarus	Trojan.Win64.Enigma

<u>If you don't have this "influxAdobeBase.dll" file, it will show this:</u>



Influx Trial Mode (3 minutes + 10 seconds for every

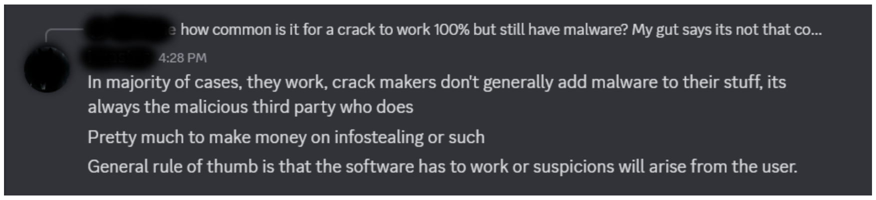
Influx for Adobe CC is not licensed, it will run in Trial Mode a
decode the first 3 minutes + 10 seconds for every minute. M
purchase on our website https://www.autokroma.com/Influx

Why this (Might) Be Safe

So why would you not just use your head and stop here? Some of it from desperation and spite, but also because **there's a decent chance it is a false-positive (labelled as malware to get people to fear it, which happens quite a bit). And also since windows defender/malwarebytes haven't detected anything else since I installed this.**

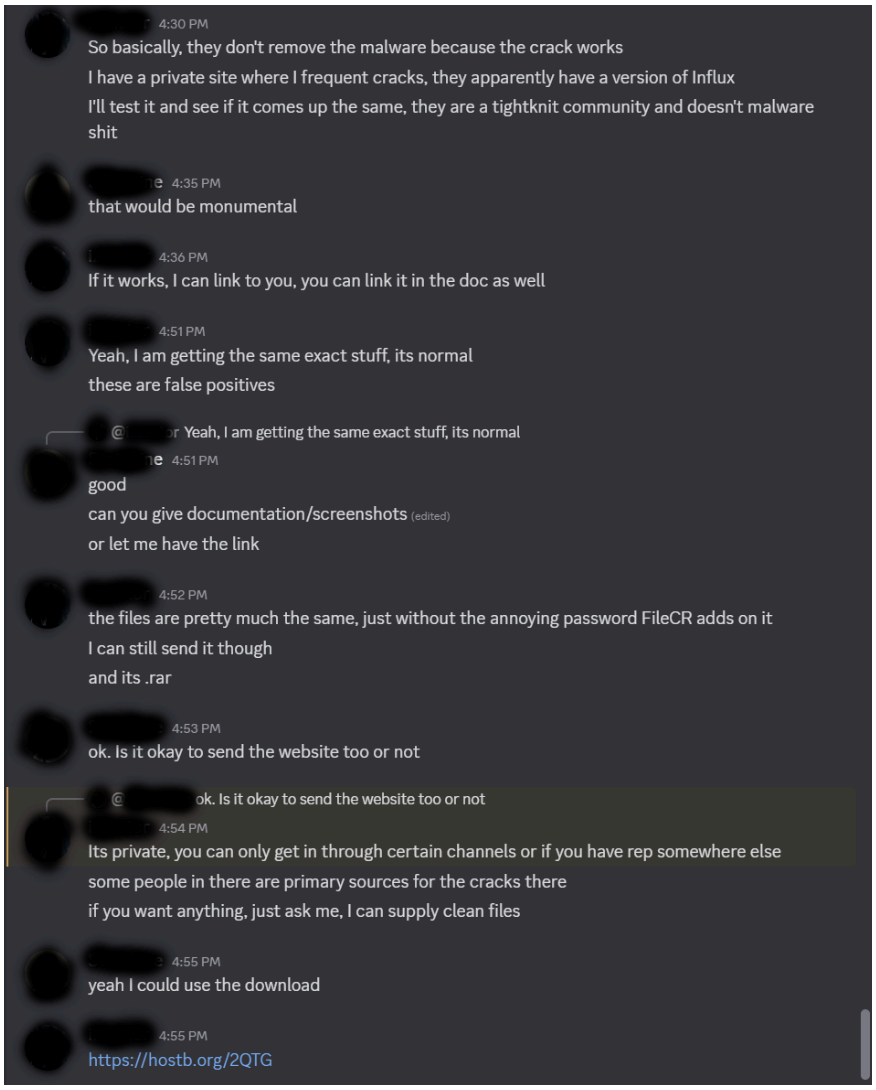
Originally I could only find a link on this mixed-reception crack site FileCR, but I asked my friend about this who has more experience with cracks. He checked his private crack community and found another link, which happened to look nearly identical to the FileCR crack (with the same virus detections).

This is as close to reassurance that it's a false positive as I can get, take it as you will. I have nothing to gain by fabricating any of this, I was going to exclude this option here until he helped calm my suspicions.



[redacted] how common is it for a crack to work 100% but still have malware? My gut says its not that co...

[redacted] 4:28 PM
In majority of cases, they work, crack makers don't generally add malware to their stuff, its always the malicious third party who does
Pretty much to make money on infostealing or such
General rule of thumb is that the software has to work or suspicions will arise from the user.



[redacted] 4:30 PM
So basically, they don't remove the malware because the crack works
I have a private site where I frequent cracks, they apparently have a version of Influx
I'll test it and see if it comes up the same, they are a tightknit community and doesn't malware shit

[redacted] 4:35 PM
that would be monumental

[redacted] 4:36 PM
If it works, I can link to you, you can link it in the doc as well

[redacted] 4:51 PM
Yeah, I am getting the same exact stuff, its normal
these are false positives

@[redacted] Yeah, I am getting the same exact stuff, its normal
[redacted] 4:51 PM
good
can you give documentation/screenshots (edited)
or let me have the link

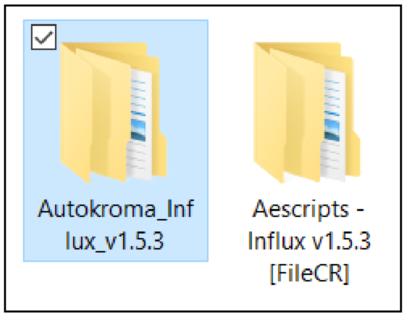
[redacted] 4:52 PM
the files are pretty much the same, just without the annoying password FileCR adds on it
I can still send it though
and its .rar

[redacted] 4:53 PM
ok. Is it okay to send the website too or not

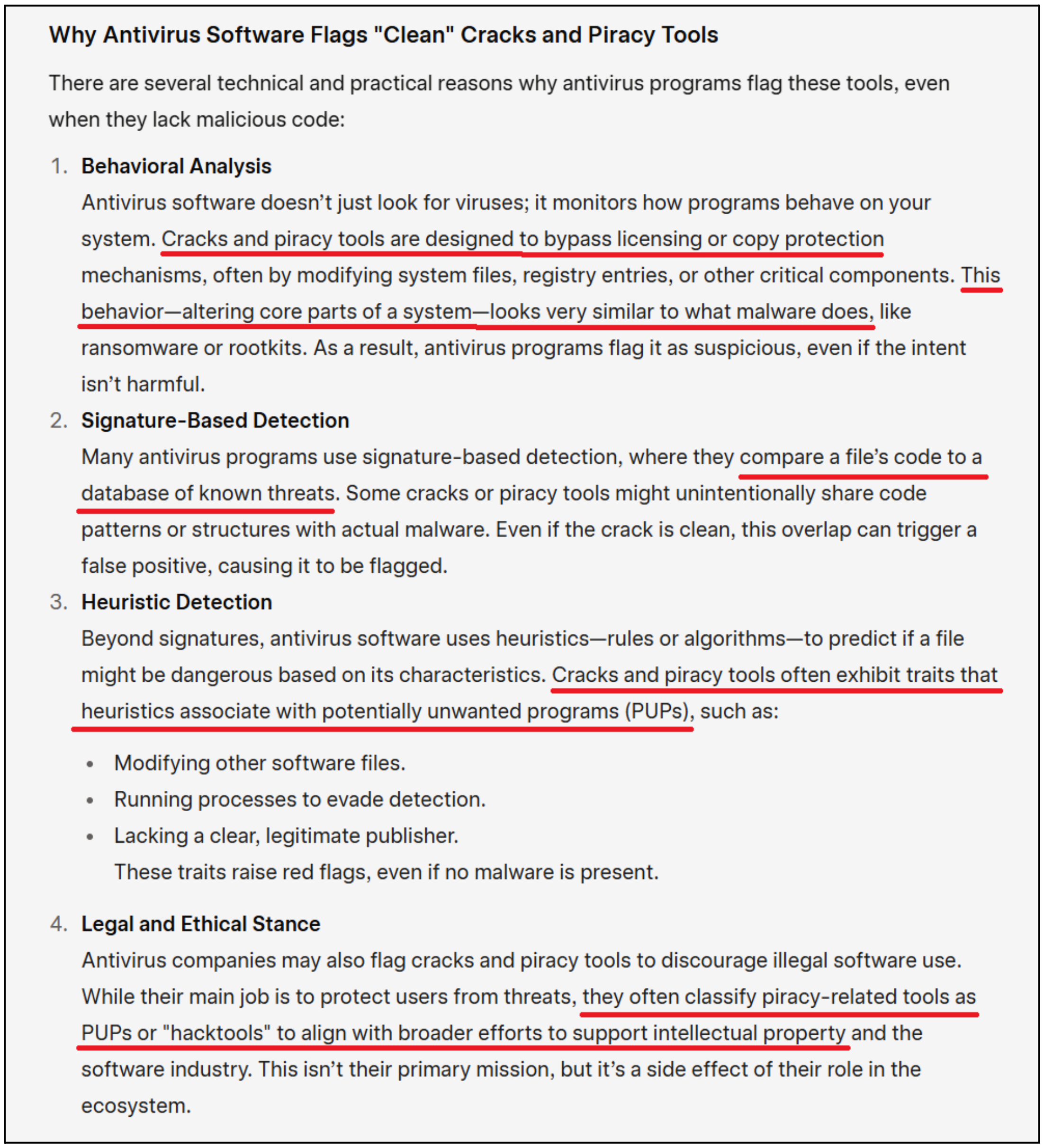
@[redacted] ok. Is it okay to send the website too or not
[redacted] 4:54 PM
Its private, you can only get in through certain channels or if you have rep somewhere else
some people in there are primary sources for the cracks there
if you want anything, just ask me, I can supply clean files

[redacted] 4:55 PM
yeah I could use the download

[redacted] 4:55 PM
https://hostb.org/2QTG



This version of the crack looks almost identical, except for the folder name and lack of a password file.



Why Antivirus Software Flags "Clean" Cracks and Piracy Tools

There are several technical and practical reasons why antivirus programs flag these tools, even when they lack malicious code:

1. **Behavioral Analysis**
   Antivirus software doesn't just look for viruses; it monitors how programs behave on your system. Cracks and piracy tools are designed to bypass licensing or copy protection mechanisms, often by modifying system files, registry entries, or other critical components. This behavior—altering core parts of a system—looks very similar to what malware does, like ransomware or rootkits. As a result, antivirus programs flag it as suspicious, even if the intent isn't harmful.
2. **Signature-Based Detection**
   Many antivirus programs use signature-based detection, where they compare a file's code to a database of known threats. Some cracks or piracy tools might unintentionally share code patterns or structures with actual malware. Even if the crack is clean, this overlap can trigger a false positive, causing it to be flagged.
3. **Heuristic Detection**
   Beyond signatures, antivirus software uses heuristics—rules or algorithms—to predict if a file might be dangerous based on its characteristics. Cracks and piracy tools often exhibit traits that heuristics associate with potentially unwanted programs (PUPs), such as:
   - Modifying other software files.
   - Running processes to evade detection.
   - Lacking a clear, legitimate publisher.
     These traits raise red flags, even if no malware is present.
4. **Legal and Ethical Stance**
   Antivirus companies may also flag cracks and piracy tools to discourage illegal software use. While their main job is to protect users from threats, they often classify piracy-related tools as PUPs or "hacktools" to align with broader efforts to support intellectual property and the software industry. This isn't their primary mission, but it's a side effect of their role in the ecosystem.

(this is why you must only trust well-known cracks, or ones from reputable communities)

MalwareBytes still finds several files and registry edits it labels as Malware/Trojans, which are unrelated to the original 5 files for the plugin. It's possible these are normal for the crack.

If you want to be safe, you can just run malwarebytes, quarantine the files and registry edits, and it will work as normal.

Quarantining the files doesn't destroy the program. I'm not sure what this implies.

I have yet to confirm this wasn't some malware I somehow had lying around on my PC (less likely, but I will test it inside a virtual machine.)

Download: **https://files.catbox.moe/uhqxrw.7z**

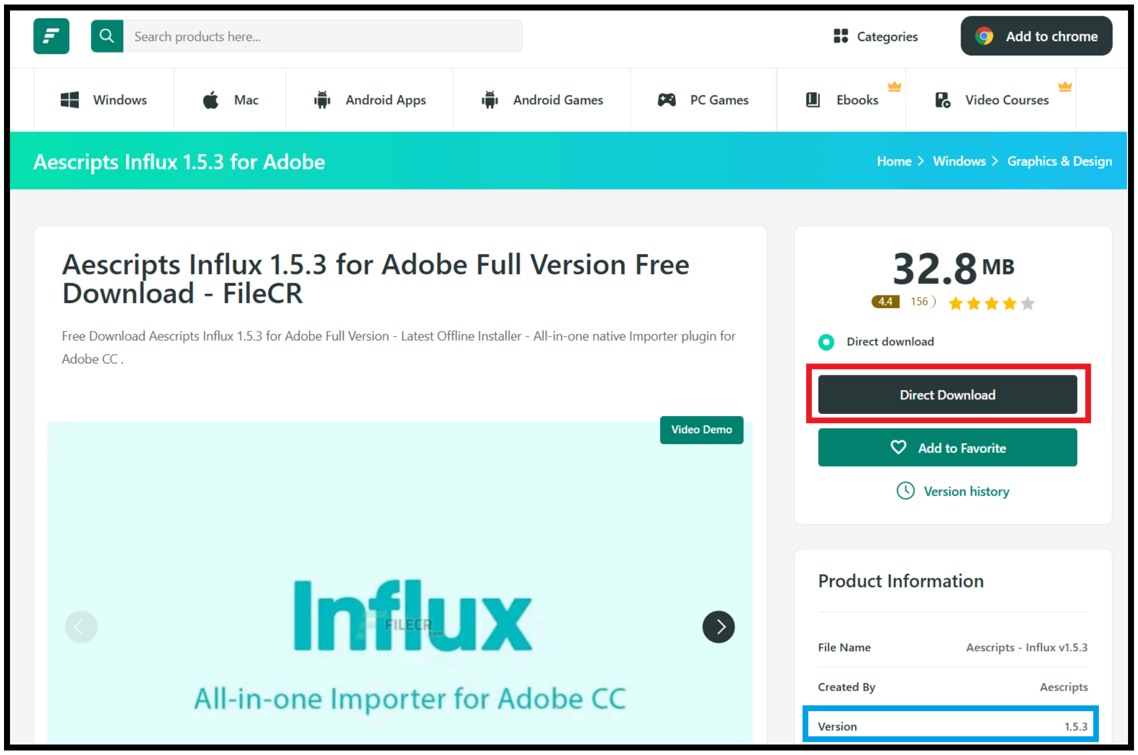
Here is the original FileCR download, it doesn't seem to be any different than the other link, but I would personally advise against using this one because FileCR has mixed reception on sites like reddit.

https://filecr.com/windows/aescripts-influx-for-after-effects/

(There are also older versions listed if you don't want to use Adobe 2025, but it's a very easy installer to use with the m0nkrus crack. Adobe 2025 also has AI transcription with a plugin m0nkrus also offers.)

1. **[FileCR] Download the correct version for Adobe.**

   If you have (or cracked) Adobe 2025, you can click "Direct Download" on the right hand side of the page.



Aescripts Influx 1.5.3 for Adobe

Aescripts Influx 1.5.3 for Adobe Full Version Free Download - FileCR

Free Download Aescripts Influx 1.5.3 for Adobe Full Version - Latest Offline Installer - All-in-one native Importer plugin for Adobe CC .

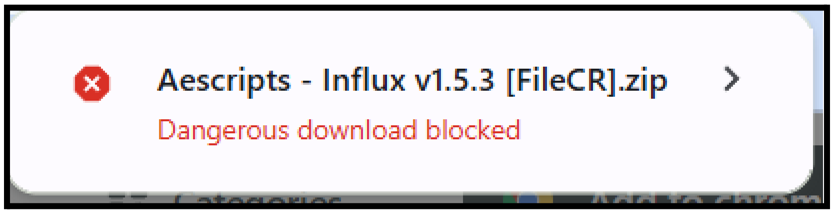
32.8 MB

4.4 156)

Product Information

File Name	Aescripts - Influx v1.5.3
Created By	Aescripts
Version	1.5.3

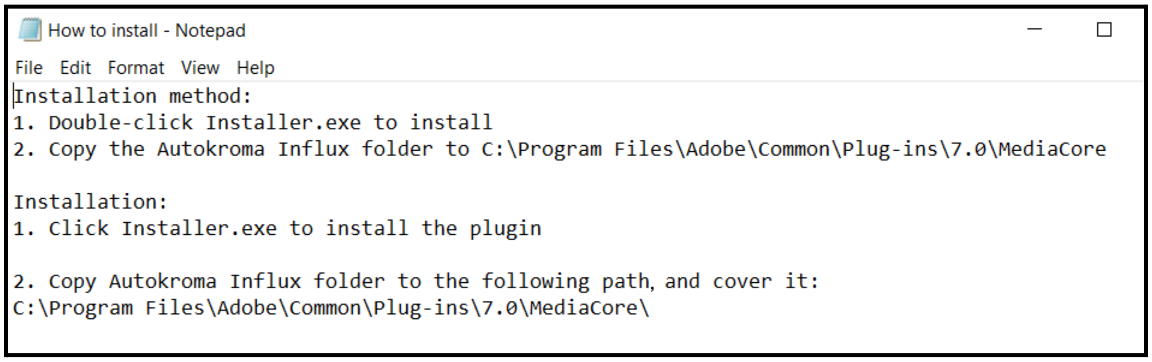
If you very well may see this error too. Crazy as it is, this may be another false-positive.



Aescripts - Influx v1.5.3 [FileCR].zip
Dangerous download blocked

If you didn't use FileCR, this step is skipped.

2. **Install Influx via the original installer, located inside the crack folder**



How to install - Notepad

File Edit Format View Help

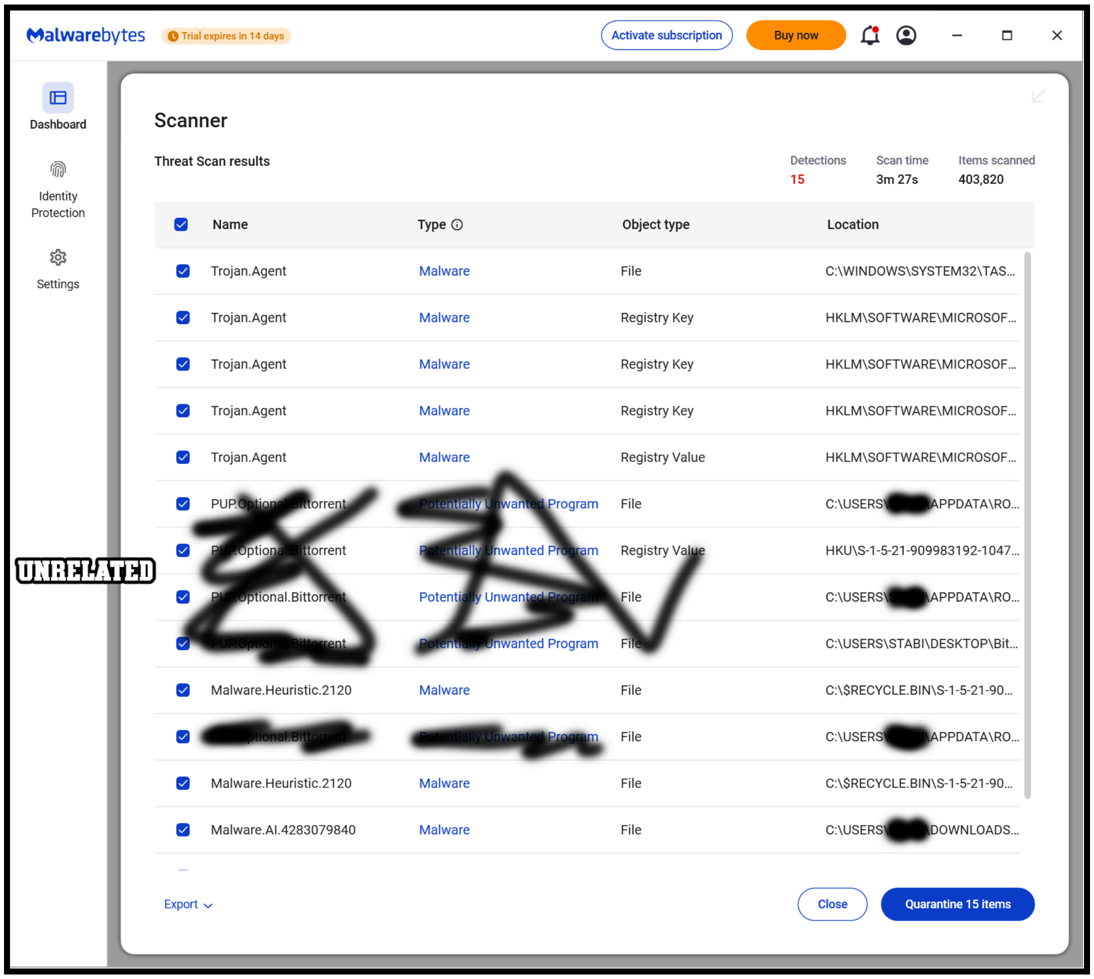
```
Installation method:
1. Double-click Installer.exe to install
2. Copy the Autokroma Influx folder to C:\Program Files\Adobe\Common\Plug-ins\7.0\MediaCore

Installation:
1. Click Installer.exe to install the plugin

2. Copy Autokroma Influx folder to the following path, and cover it:
C:\Program Files\Adobe\Common\Plug-ins\7.0\MediaCore\
```

3. **Open this folder located in this directory**

   C:\Program Files\Adobe\Common\Plug-ins\7.0\MediaCore\Autokroma Influx

4. **Replace the 5 files from the Influx crack directory into the official plugin's directory.**

Given the crack works and it might be a false-positive, **MalwareBytes finding all these trojan.Agents and registry edits is worrying**. As I warned, there's still a chance it's real malware. In my opinion it is worth saving 90$ to take this risk for a bullshit problem nobody should have to pay to solve.



Scanner

Threat Scan results

Detections 15 | Scan time 3m 27s | Items scanned 403,820

Name	Type	Object type	Location
Trojan.Agent	Malware	File	C:\WINDOWS\SYSTEM32\TAS...
Trojan.Agent	Malware	Registry Key	HKLM\SOFTWARE\MICROSOF...
Trojan.Agent	Malware	Registry Key	HKLM\SOFTWARE\MICROSOF...
Trojan.Agent	Malware	Registry Key	HKLM\SOFTWARE\MICROSOF...
Trojan.Agent	Malware	Registry Value	HKLM\SOFTWARE\MICROSOF...
[redacted]	[redacted]	File	C:\USERS\[redacted]\APPDATA\RO...
[redacted]	[redacted]	Registry Value	HKU\S-1-5-21-909983192-1047...
[redacted]	[redacted]	File	C:\USERS\[redacted]\APPDATA\RO...
[redacted]	[redacted]	File	C:\USERS\STABI\DESKTOP\Bit...
Malware.Heuristic.2120	Malware	File	C:\$RECYCLE.BIN\S-1-5-21-90...
[redacted]	[redacted]	File	C:\USERS\[redacted]\APPDATA\RO...
Malware.Heuristic.2120	Malware	File	C:\$RECYCLE.BIN\S-1-5-21-90...
Malware.AI.4283079840	Malware	File	C:\USERS\[redacted]\DOWNLOADS...

But, after removing all these potential malware files, somehow **the program still works** (indicating the original plugin files aren't exactly the problem). I haven't seen any new malware traces since installing it here.

There's a chance some of this might be unrelated to this, as I haven't run MalwareBytes in awhile. But you can still see a few results related to Influx files, not just the registry edits.

<u>The Old Manual Solution (Definitely Safe, but Tedious)</u>

If you're not willing to install something detected as Malware (yes, it sucks we have to do this):

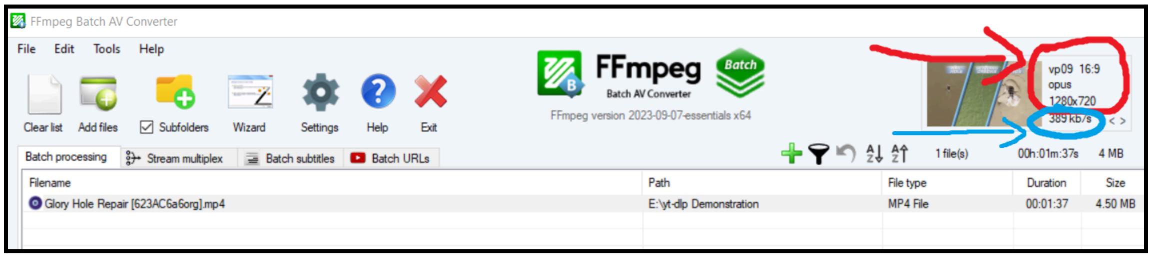
You have to manually re-encode your videos to AVC (h264) format to import into Sony Vegas (or Adobe if you didn't install the crack). OPUS must also be eliminated as the audio format (converted to .aac), but this happens automatically.

Luckily, this is easy, but may be tedious and time consuming depending on PC hardware. If you're doing it consistently as a video editor, it will be even worse.

1. Install FFmpeg Batch AV Converter (separate thing from FFmpeg.exe, **way** more useful):

   https://sourceforge.net/projects/ffmpeg-batch/files/FFBatch_AV_Converter_3.0.7_64bit.exe/download

2. Drag the video file into the program.



FFmpeg version 2023-09-07-essentials x64

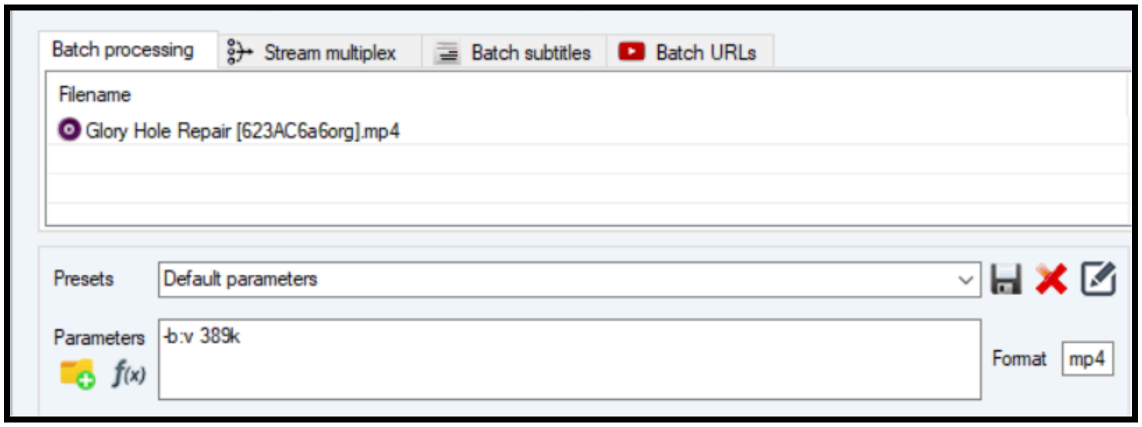
vp09 16:9
opus
1280x720
389 kb/s

1 file(s) 00h:01m:37s 4 MB

Filename	Path	File type	Duration	Size
Glory Hole Repair [623AC6a6org].mp4	E:\yt-dlp Demonstration	MP4 File	00:01:37	4.50 MB

3. Look at the bitrate from the top-right corner, where it says vp09 and opus. Here, it says 389kb/s. Just remember whatever number it says.

4. Change the 'parameters' section to: -b:v (insert bitrate number + k)



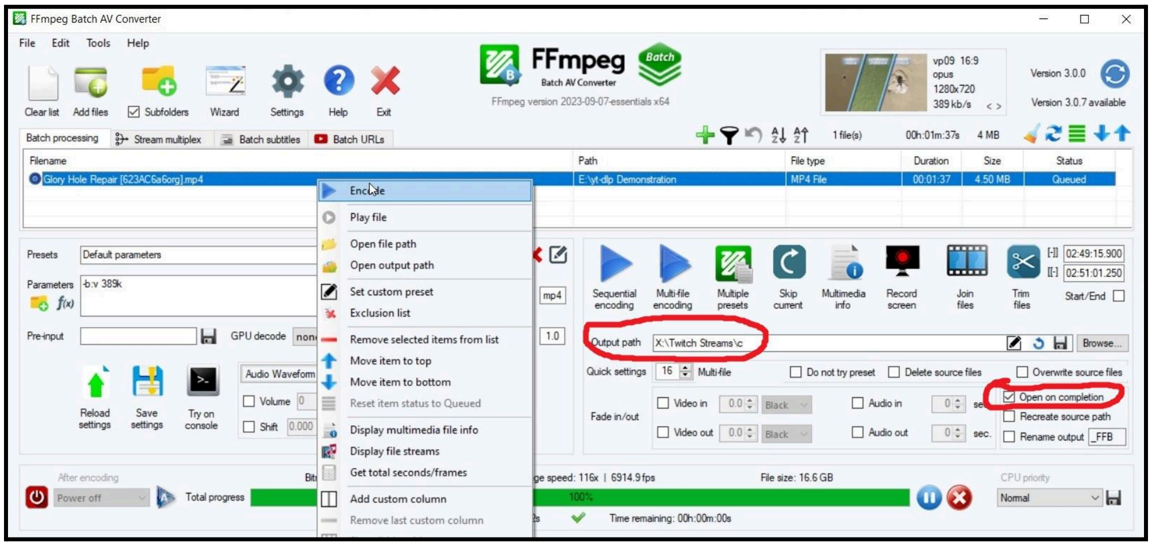
Glory Hole Repair [623AC6a6org].mp4

Presets: Default parameters

Parameters: -b:v 389k

Format: mp4

5. Right click the video, and click encode. Check “Open on Completion” if you want it to open the folder where the re-encoded video was made.



FFmpeg version 2023-09-07-essentials x64

vp09 16:9 opus 1280x720 389 kb/s

1 file(s) 00h:01m:37s 4 MB

Filename	Path	File type	Duration	Size	Status
Glory Hole Repair [623AC6a6org].mp4	E:\yt-dlp Demonstration	MP4 File	00:01:37	4.50 MB	Queued

Presets: Default parameters

Parameters: -b:v 389k

Format: mp4

Pre-input: GPU decode: none

Output path: X:\Twitch Streams\c

Quick settings: 16 Multi-file

☐ Do not try preset ☐ Delete source files ☐ Overwrite source files

☑ Open on completion

☐ Recreate source path

☐ Rename output _FFB

Fade in/out: ☐ Video in 0.0 Black ☐ Video out 0.0 Black ☐ Audio in 0 sec. ☐ Audio out 0 sec.

[-] 02:49:15.900 [-] 02:51:01.250

Audio Waveform ☐ Volume 0 ☐ Shift 0.000

After encoding: Power off

Average speed: 116x | 6914.9 fps File size: 16.6 GB 100%

Time remaining: 00h:00m:00s

CPU priority: Normal

6. You now have a video-editor compatible video. You can do this for multiple files at the same time, but they will have the same bitrate on all of them.

FFmpeg AV Batch Convert is also **very** useful for rapidly clipping segments of videos using timestamps, *this is covered more in my Easy Clipping document (top of this document).*

If you have the video file downloaded, like with yt-dlp, it's a very fast option especially for finding relevant sections of videos you want to edit down later.

This is an extremely underrated technique for people who edit content, like for clip channels. If I was a youtuber I would spread this info around like gospel.

Paired with a super-fast video player like mpv, you can scrub through hours of content in minutes, while still getting the gist of what's being said, and turn it into clips to edit down later.

Mass-Convert VP9 to AVC (Python Script)

The speed of this is, again, limited by your PC hardware. It's no faster than FFmpeg Batch Converter, it's just much more automated (***works on any amount of files at their specific maximum bitrates***).

If you have more than a few files to re-encode (like a whole playlist or entire channel), I have a script that can accomplish this for any amount of files.

(see: "Convert all VP9/OPUS Files into H264 (for video editing)" section)

You will have to install a python shell like IDLE to run the script, but it is generally quite simple.

Other Fixes for VP9 Bullshit?

Yes this is a tutorial and I'm using swears. Sincerely, fuck Google in every way possible.

If you're in the minority of people who need to convert a video from VP9, and don't use Adobe Premiere (or want to avoid the above options), these are the other options I have tried; they're not perfect so I'd advise you use the other solutions as a *fallback* at least.

None of these can download over 1080p60 quality, only vp9 can do this. If you need over 1080p, you have to either re-encode, or use the Adobe plugin.

Forcing a List of AVC Video Formats (Max 1080p60)

This is a command which uses a list of formats in descending order of quality, all the way down to a fallback option, to

--format (305/266/304/264/299/137/298/136/135/134/133/160/bestvideo)+140

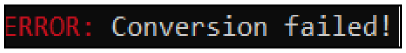
This is probably the ideal solution to this problem, if you're unwilling to install an Adobe plugin crack, or have no reason to try to download over 1080p videos.

Forcing Post-Processing Transcoding (Max 1080p60, No Embedding Thumbnail)

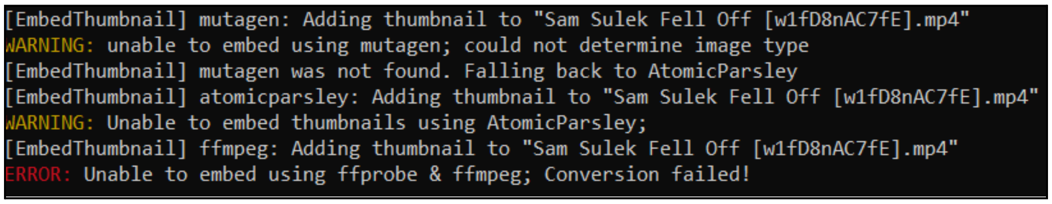
yt-dlp has post-processing arguments which can transcode a video after downloading it, but before the yt-dlp command is finished. This will be GPU-dependent.

These commands were generated by yt-dlp, so I'm not sure if it's written correctly, But the command does not work with --embed-thumbnail.

<u>If using --embed-thumbnail, it will always show this error and won't be able to merge the files</u>:

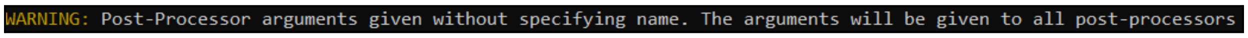


```
ERROR: Conversion failed!
```

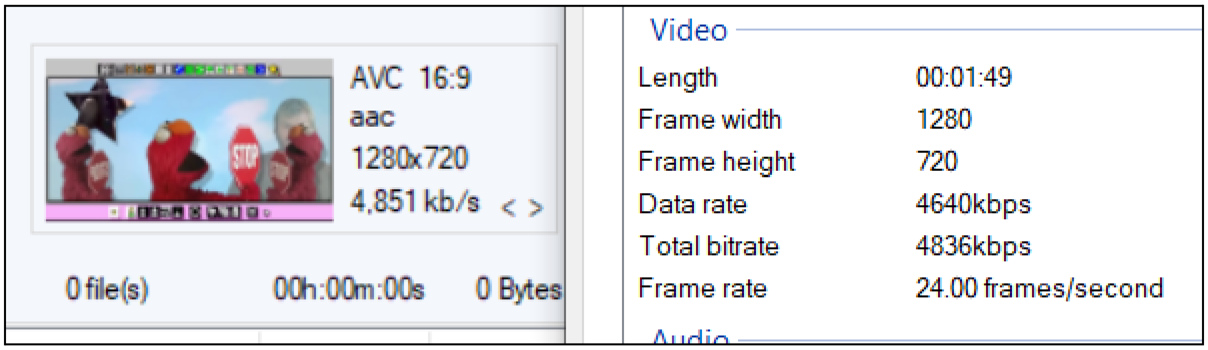


```
[EmbedThumbnail] mutagen: Adding thumbnail to "Sam Sulek Fell Off [w1fD8nAC7fE].mp4"
WARNING: unable to embed using mutagen; could not determine image type
[EmbedThumbnail] mutagen was not found. Falling back to AtomicParsley
[EmbedThumbnail] atomicparsley: Adding thumbnail to "Sam Sulek Fell Off [w1fD8nAC7fE].mp4"
WARNING: Unable to embed thumbnails using AtomicParsley;
[EmbedThumbnail] ffmpeg: Adding thumbnail to "Sam Sulek Fell Off [w1fD8nAC7fE].mp4"
ERROR: Unable to embed using ffprobe & ffmpeg; Conversion failed!
```

With these minor limitations, it should be able to convert to AVC+aac.



```
WARNING: Post-Processor arguments given without specifying name. The arguments will be given to all post-processors
```



AVC 16:9
aac
1280x720
4,851 kb/s

0 file(s) 00h:00m:00s 0 Bytes

Video

Length	00:01:49
Frame width	1280
Frame height	720
Data rate	4640kbps
Total bitrate	4836kbps
Frame rate	24.00 frames/second

Audio

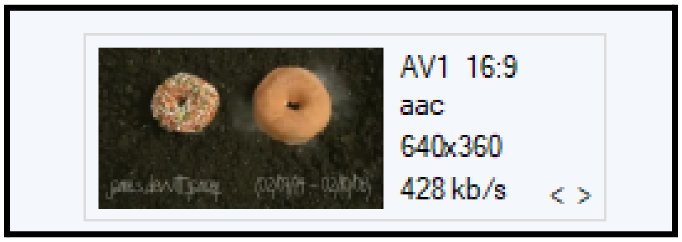
<u>Forcing Non-VP9 Downloads (Broken)</u>

This yt-dlp command may force yt-dlp to pick video codecs other than vp9 and OPUS audio.

-f "bestvideo[vcodec!=vp9][vcodec!*=av01]+bestaudio[acodec^=mp4a]"*

I have tested this once, and received an AV1 file with AAC audio which is ideal.

But I tried it on another video, and it downloaded in **extremely** low quality per usual (since there were no better options according to yt-dlp, because google destroyed them).



So this option is **very inconsistent** and you might as well not use it because it won't work all the time, and **each time you use it you'd have to verify the video didn't download at 360p (unless the video's max quality was).**

Other Useful Commands

(the first two don't download videos)

- yt-dlp -U

  (Updates the program; this is CRUCIAL especially nowadays as Youtube continues changing their API).

- yt-dlp -F [URL]
  (lists available formats for video, to use them type "yt-dlp -f [format number]").

Example: yt-dlp -f 136+251 [URL]

(136 is a video format, 251 is an audio format)

- --download-archive [FILENAME].txt

  (Saves urls of downloaded videos to a text file, so it never downloads them again).

  Make sure to save your own yt-dlp template with the correct download archive location. I normally use sticky notes on my PC for this, otherwise a Word document.

  ***Highly* Recommended**

- -a [FILENAME].txt

  (Downloads a list of URLs from a .txt file).

- --cookies C:/Users/[USER]/Downloads/cookies.txt

  (Downloads age-restricted videos, or verifies identity as a non-bot).

  Scroll down for instructions on how to use this command.

- --compat-options filename-sanitization

  (Prevents yt-dlp from using alternate unicode characters when Windows blocks them in filenames; they look unsightly to me, and often are incompatible with certain websites).

  Examples are * < > " : / \ |

  Recommended.

- -o "(%(upload_date)s) %(title)s [%(id)s]"

(Lists the upload date at the beginning of the filename (raw, in parenthesis): useful for organizing large channels in chronological order).

- -o "[%(upload_date>%Y-%m-%d)s] %(title)s [%(id)s]"

  (Lists the upload date at the beginning of the filename [separated by dashes, in brackets]: useful for organizing large channels in chronological order).

  <u>Recommended, especially if you don't save info.json files for later reference.</u>

  <u>Sorting by date can be very helpful for archivists.</u>

- -o "%(uploader)s - %(title)s [%(id)s]"

  (Lists the Channel Name at the beginning of the filename like this Channel Name - Video Title [Video ID]: useful for sorting videos by channel name, like for music or sorting into folders).

To use the cookies command, download your Youtube cookies file

1. Install this extension in Google Chrome / Brave: Get cookies.txt - Chrome Web Store (google.com)
2. *Or Microsoft edge*: Get cookies.txt - Microsoft Edge Addons
3. Open the youtube homepage (while logged in with your account).
4. Click the extension in top right, then click the "Export All Cookies" button.
5. Copy everything in "youtube.com_cookies.txt" and paste into the "cookies.txt" file the command uses. Or just rename the file cookies.txt

6. Add the cookies command to your full yt-dlp command and *make sure* to have the correct file path listed in the command.

For More Commands

Scroll down to "General Options" or Ctrl-F it on this page: https://github.com/yt-dlp/yt-dlp

SPECIAL TOOLS

All of these were vibe-coded with AI, but it doesn't make them less useful/functional. It makes them easily customizable, and anyone can make them with enough prompt context. You can easily alter any of these for your own use cases. AI is very good at basic scripts, almost too good.
But as soon as you start to have to troubleshoot, expect torture, ChatGPT for instance is pathetically bad at fixing it's own errors even if explained correctly. It may take dozens of hours to get a really complex script, even if it's easy on paper.

Before trying any of these, you'll need to:

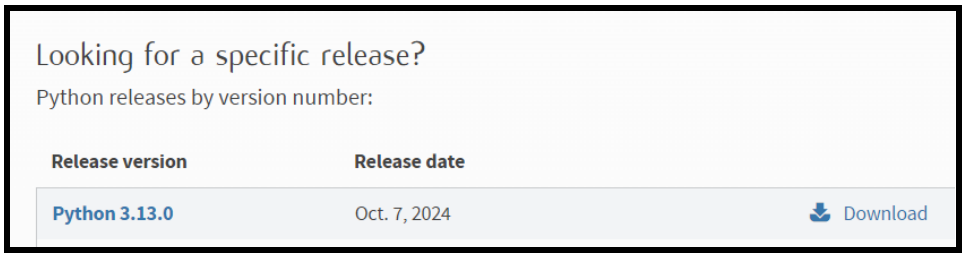
- **Have Python IDLE or a similar shell installed (I used IDLE so that's what I'll describe).** It's really easy to set up and use for script kiddies like myself.

- **Install GitBash (not required on linux) to install packages Python will use.** For any tool, the required packages will be given so you can run the script after you install them.

- **Learn some of the shortcuts to use them quickly.**

For Python IDLE:

1. Install Python IDLE. With it you'll be able to run .py script files. **This will come in handy very much.**



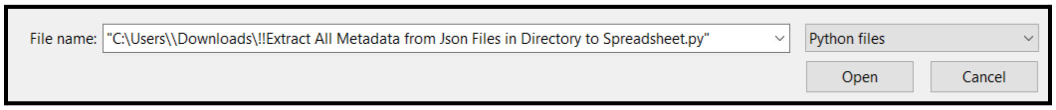
Looking for a specific release?

Python releases by version number:

Release version	Release date	
Python 3.13.0	Oct. 7, 2024	Download

2. **Learn how to open/run Python scripts.**

    - Open them either by opening the Python IDLE shell and pressing CTRL-O, then CTRL-C on the script in the folder (same as copying the file), and pressing CTRL-V inside the File name box.



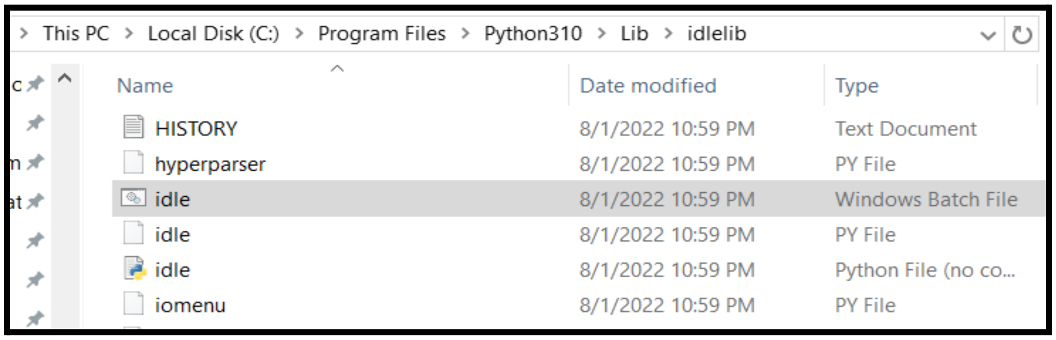
File name: "C:\Users\\Downloads\!!Extract All Metadata from Json Files in Directory to Spreadsheet.py"

Python files

    - Or by setting IDLE.exe as the default program to open .py files. This is more complicated than it should be; setting python.exe or pythonw.exe as default will not open the script in IDLE if the file is double-clicked, at least in my experience.

      You'll have to find the install directory of python, it should be inside a similar folder to this:
      C:\Program Files\Python310\Lib\idlelib

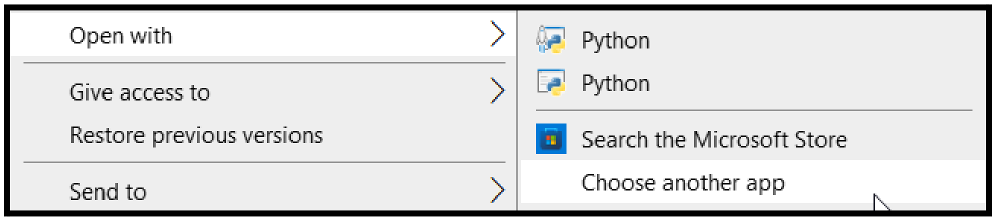
      Find **idle.bat**.

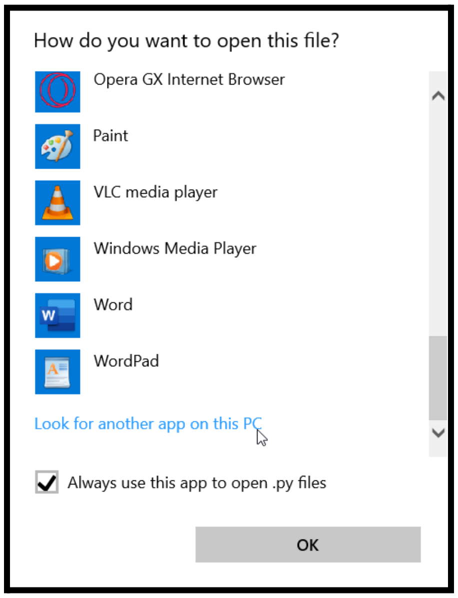


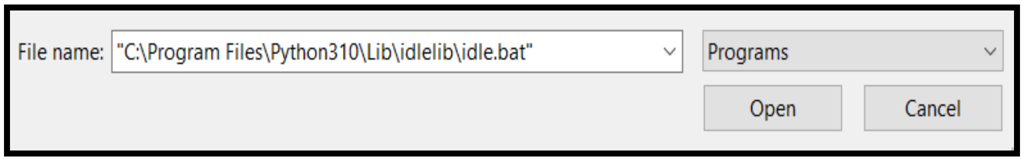
This PC > Local Disk (C:) > Program Files > Python310 > Lib > idlelib

Name	Date modified	Type
HISTORY	8/1/2022 10:59 PM	Text Document
hyperparser	8/1/2022 10:59 PM	PY File
idle	8/1/2022 10:59 PM	Windows Batch File
idle	8/1/2022 10:59 PM	PY File
idle	8/1/2022 10:59 PM	Python File (no co...
iomenu	8/1/2022 10:59 PM	PY File

- Right click the script and Choose another app, click "More Apps" and scroll until "Look for another app on this PC". Make sure the box is checked, then click and paste the location of idle.bat into the File name box, using the same command as copying a file.




How do you want to open this file?
Opera GX Internet Browser
Paint
VLC media player
Windows Media Player
Word
WordPad
Look for another app on this PC
Always use this app to open .py files
OK



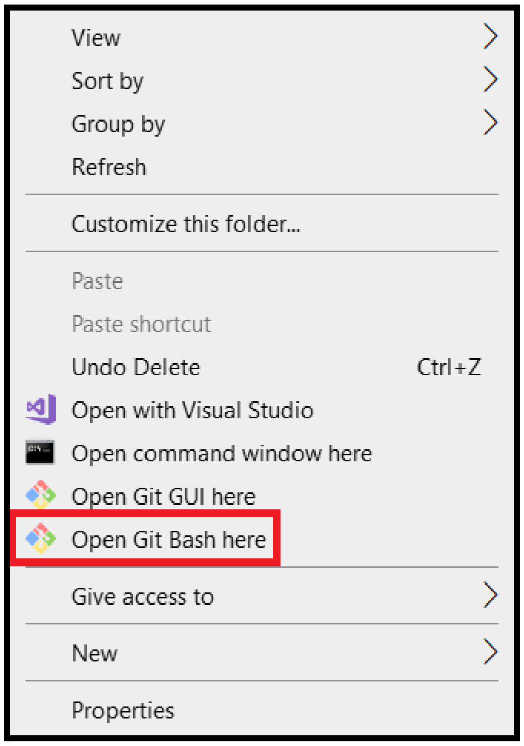
File name: "C:\Program Files\Python310\Lib\idlelib\idle.bat"

Programs

3. Open a script in IDLE just by double-clicking the .py file, and then press F5 on the keyboard or select “Run Module” from the Run option on the toolbar.

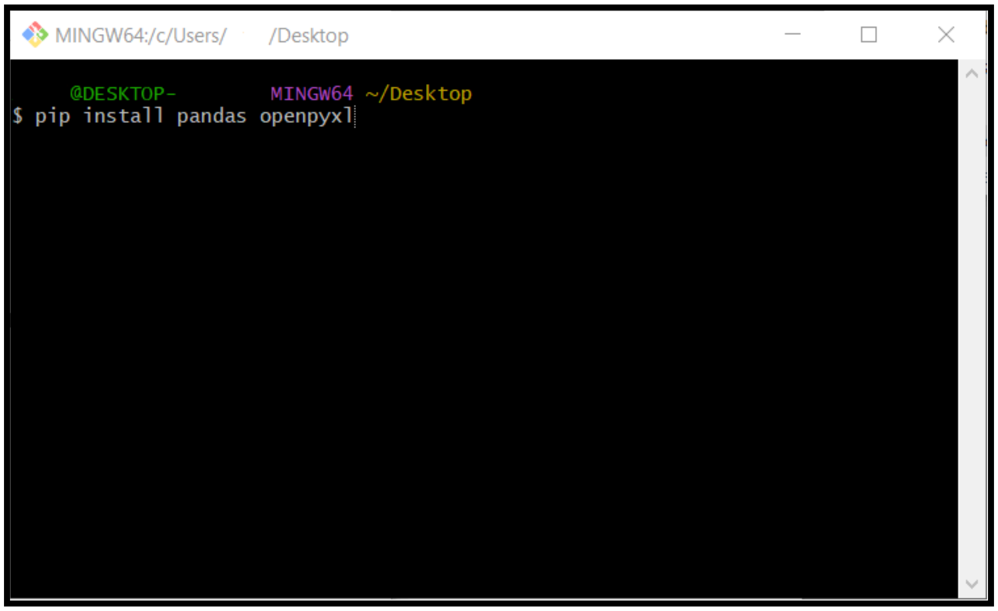
<u>**For Gitbash:**</u>

1. Install GitBash (unless you’re on Linux). Then open the terminal by right clicking in any folder and clicking “Open Git Bash here.”



2. To install any required packages, just run this command:

   pip install [INSERT NAME OF PACKAGE(s)]



```
    @DESKTOP-        MINGW64 ~/Desktop
$ pip install pandas openpyxl
```

24/7 Stream Archive Loops

There's also a separate link to this earlier in the document, but this section will have much more detail.

It's very easy to set up a looping script on Windows/Linux which will automatically archives Youtube/Twitch streams of any channel, as well as from other sites. It requires even less understanding of yt-dlp to use, since it just runs out the box.

If you're extra serious about this, you could set these up on a Debian Linux PC / server, or a Raspberry Pi so you don't have to think about it, until you want to upload / use the files.

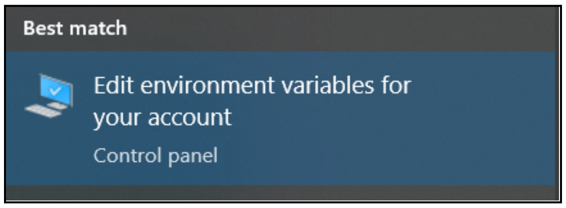
- It also works for Youtube live chat, which can be archived on its own using (see: "RENDER YT-DLP LIVE-CHAT" section)

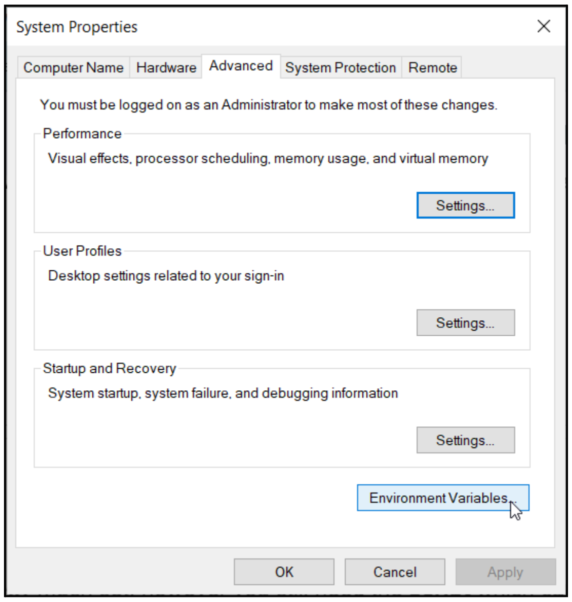
- And it works for youtube channels themselves, to download the videos as soon as they're posted (if you archive comments, this will not save any).

Windows

1. **Place the latest yt-dlp.exe file into your directory of choice (downloaded from here).**

   You can also set yt-dlp.exe in your PATH, so it can be referenced anywhere from your PC.



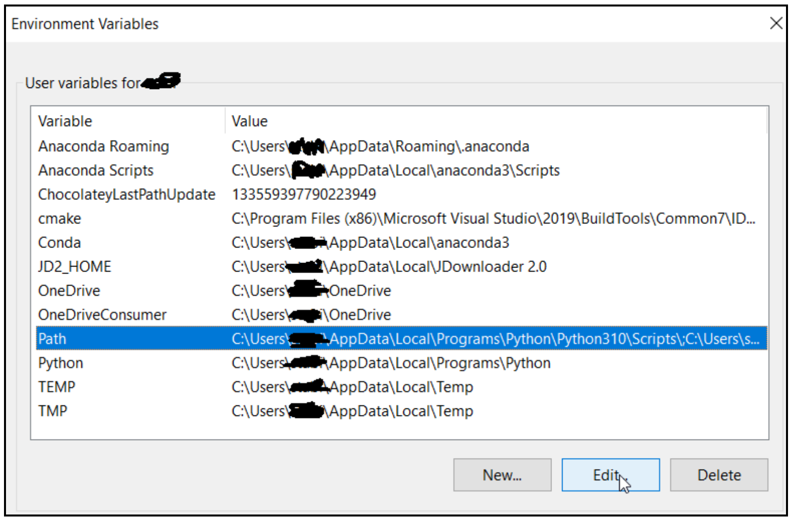


You must be logged on as an Administrator to make most of these changes.

Performance
Visual effects, processor scheduling, memory usage, and virtual memory

User Profiles
Desktop settings related to your sign-in

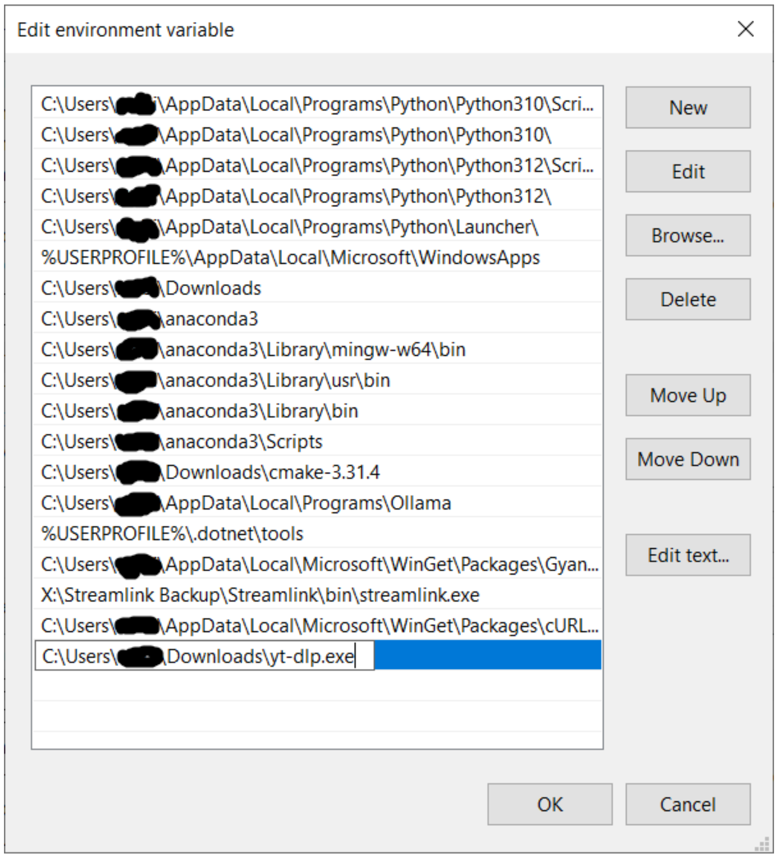
Startup and Recovery
System startup, system failure, and debugging information


Environment Variables

User variables for [redacted]

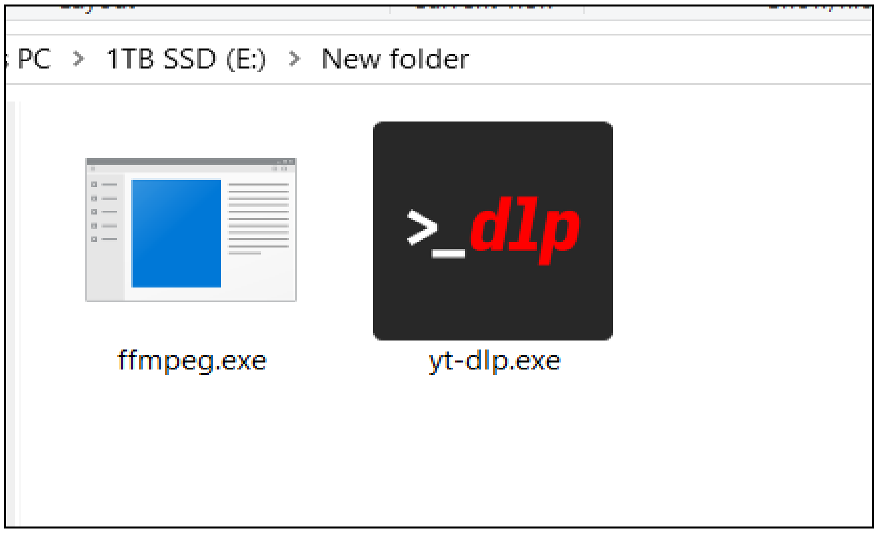
Variable	Value
Anaconda Roaming	C:\Users\[redacted]\AppData\Roaming\.anaconda
Anaconda Scripts	C:\Users\[redacted]\AppData\Local\anaconda3\Scripts
ChocolateyLastPathUpdate	133559397790223949
cmake	C:\Program Files (x86)\Microsoft Visual Studio\2019\BuildTools\Common7\ID...
Conda	C:\Users\[redacted]\AppData\Local\anaconda3
JD2_HOME	C:\Users\[redacted]\AppData\Local\JDownloader 2.0
OneDrive	C:\Users\[redacted]\OneDrive
OneDriveConsumer	C:\Users\[redacted]\OneDrive
Path	C:\Users\[redacted]\AppData\Local\Programs\Python\Python310\Scripts\;C:\Users\s...
Python	C:\Users\[redacted]\AppData\Local\Programs\Python
TEMP	C:\Users\[redacted]\AppData\Local\Temp
TMP	C:\Users\[redacted]\AppData\Local\Temp

Set the full directory where yt-dlp is located as a new PATH variable (which you will continually update as needed).


Edit environment variable

```
C:\Users\[redacted]\AppData\Local\Programs\Python\Python310\Scri...
C:\Users\[redacted]\AppData\Local\Programs\Python\Python310\
C:\Users\[redacted]\AppData\Local\Programs\Python\Python312\Scri...
C:\Users\[redacted]\AppData\Local\Programs\Python\Python312\
C:\Users\[redacted]\AppData\Local\Programs\Python\Launcher\
%USERPROFILE%\AppData\Local\Microsoft\WindowsApps
C:\Users\[redacted]\Downloads
C:\Users\[redacted]\anaconda3
C:\Users\[redacted]\anaconda3\Library\mingw-w64\bin
C:\Users\[redacted]\anaconda3\Library\usr\bin
C:\Users\[redacted]\anaconda3\Library\bin
C:\Users\[redacted]\anaconda3\Scripts
C:\Users\[redacted]\Downloads\cmake-3.31.4
C:\Users\[redacted]\AppData\Local\Programs\Ollama
%USERPROFILE%\.dotnet\tools
C:\Users\[redacted]\AppData\Local\Microsoft\WinGet\Packages\Gyan...
X:\Streamlink Backup\Streamlink\bin\streamlink.exe
C:\Users\[redacted]\AppData\Local\Microsoft\WinGet\Packages\cURL...
C:\Users\[redacted]\Downloads\yt-dlp.exe
```

2. **Place ffmpeg.exe into your directory of choice (or add it to PATH). If finding the ffmpeg.exe is complex, here's a direct download.** *(don't run it, like yt-dlp it's not an installer)*



3. **Put both .exe's in a folder, then save the section of text at the bottom (starting with the @, scroll down a bit for that) to a text document .txt file IN THAT FOLDER.**

4. **Then replace the filler text with the streamer you're trying to archive.**

   Typically, you want to use the channel URL in this format

   https://www.youtube.com/@IShowSpeed/live

   This should work almost always, but if for some reason you see an error, you can use:

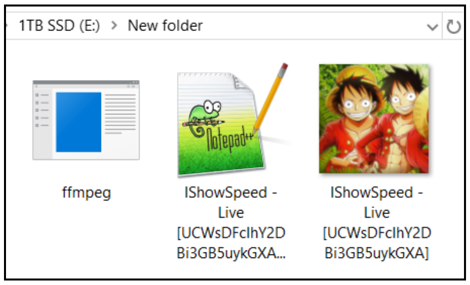
   https://www.youtube.com/@IShowSpeed/streams

   Or just the channel handle:

   https://www.youtube.com/@IShowSpeed

   Using the handle isn't perfect either, since people change their channel handles from time to time, breaking the loop.

   Ideally, you will find the channel URL, which can be found in a json file when you try to download the entire channel of a youtuber.

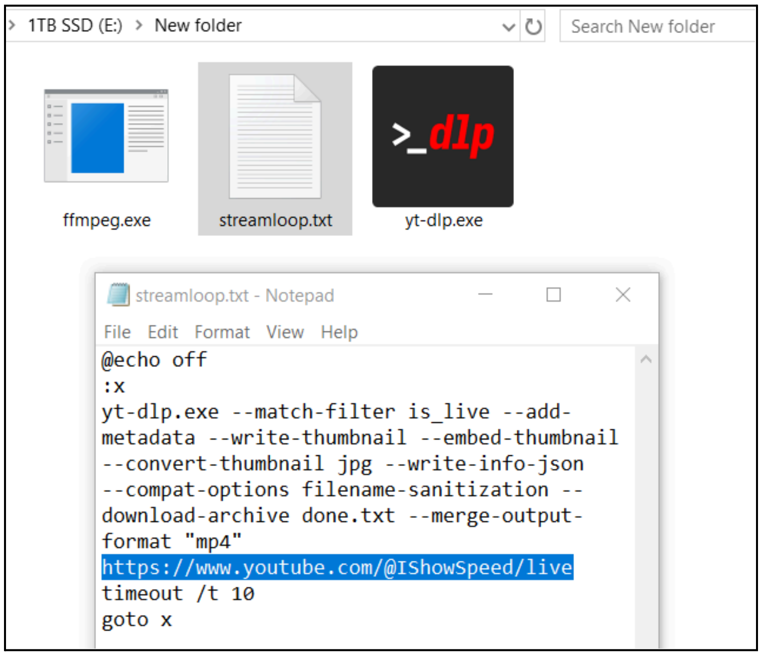

1TB SSD (E:) > New folder
ffmpeg
IShowSpeed - Live [UCWsDFcIhY2DBi3GB5uykGXA...
IShowSpeed - Live [UCWsDFcIhY2DBi3GB5uykGXA]

For example, for @IShowSpeed, the channel URL is:

https://www.youtube.com/channel/UCWsDFcIhY2DBi3GB5uykGXA

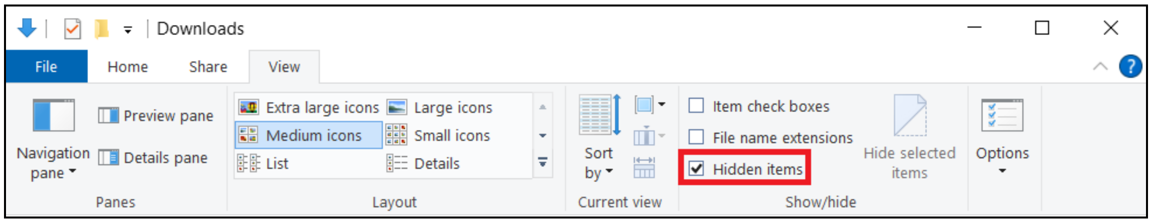
Then you can add the live suffix to it. This will work universally.

https://www.youtube.com/channel/UCWsDFcIhY2DBi3GB5uykGXA/live

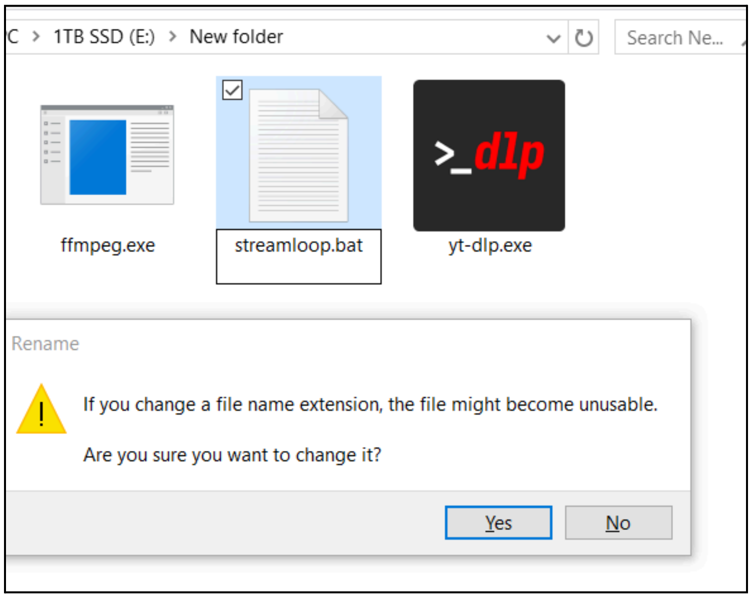


```
@echo off
:x
yt-dlp.exe --match-filter is_live --add-metadata --write-thumbnail --embed-thumbnail --convert-thumbnail jpg --write-info-json --compat-options filename-sanitization --download-archive done.txt --merge-output-format "mp4" https://www.youtube.com/@IShowSpeed/live
timeout /t 10
goto x
```

5. **Make sure file extensions are enabled in the view section, then rename the .txt part of the file to .bat. This converts it.**



☐ Item check boxes
☐ File name extensions
☑ Hidden items



Rename

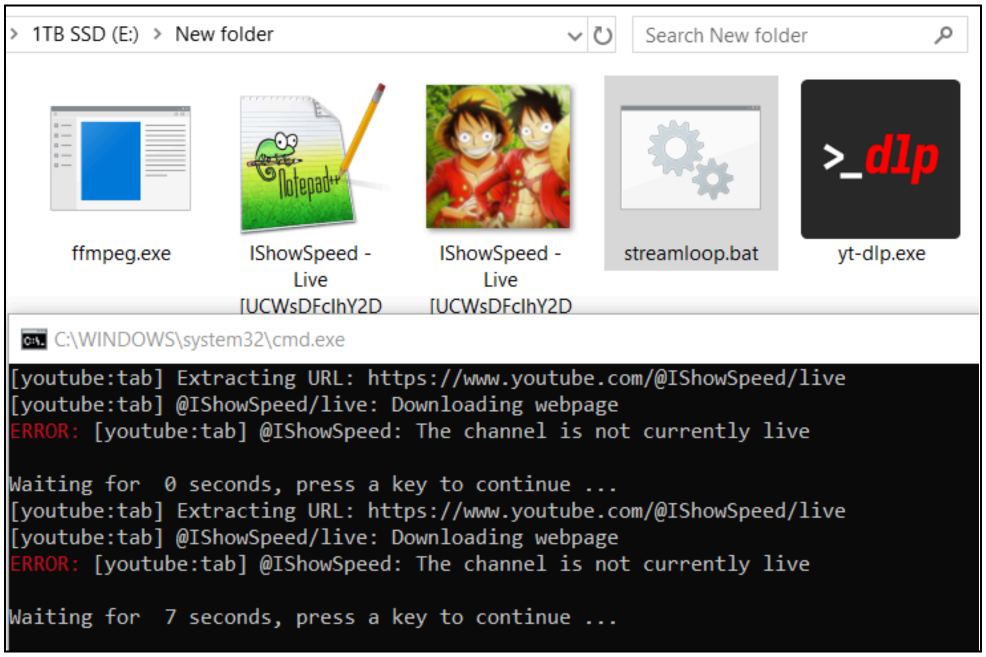
If you change a file name extension, the file might become unusable.

Are you sure you want to change it?

Yes No

6. **Simply double-click and run it when you're trying to archive streams (which can be 24/7 if your PC stays on). Done.**

   It will run in a loop until it finds a stream.


1TB SSD (E:) > New folder
Search New folder
ffmpeg.exe
IShowSpeed - Live [UCWsDFcIhY2D
IShowSpeed - Live [UCWsDFcIhY2D
streamloop.bat
yt-dlp.exe
>_dlp
C:\WINDOWS\system32\cmd.exe

```
[youtube:tab] Extracting URL: https://www.youtube.com/@IShowSpeed/live
[youtube:tab] @IShowSpeed/live: Downloading webpage
ERROR: [youtube:tab] @IShowSpeed: The channel is not currently live

Waiting for  0 seconds, press a key to continue ...
[youtube:tab] Extracting URL: https://www.youtube.com/@IShowSpeed/live
[youtube:tab] @IShowSpeed/live: Downloading webpage
ERROR: [youtube:tab] @IShowSpeed: The channel is not currently live

Waiting for  7 seconds, press a key to continue ...
```

NOTE: yt-dlp will not save streams over 6 hours.

There is a hard-cap to the length of the livestream file (for some reason), so if they stream more than 6 hours your file might be split in two if it doesn't just end abruptly (**and you'll have to join them with a program like FFmpeg Batch AV Convert**).

^^Streamlink doesn't have this issue, read further for info on that.

Windows .bat Loop Template

@echo off
:x
yt-dlp.exe --match-filter is_live --add-metadata --write-thumbnail --embed-thumbnail --convert-thumbnail jpg --write-info-json --compat-options filename-sanitization --download-archive done.txt --merge-output-format "mp4" ***[REPLACE WITH URL OF LIVESTREAMER, THIS CAN BE YOUTUBE, TWITCH, ETC.]***
timeout /t 20
goto x

Save this as a text file, and rename the file extension to .bat.

Warning: running multiple of these may increase your idle CPU core temperatures.

Even if the script takes up a **very low** CPU % Usage, I have noticed (on a powerful AMD Ryzen 9950x) that running only two of these increases my idle temperatures by 10°C (~50 → ~60)

I was running 8 at the time, and the temperatures were elevated by a whopping 20°C (~50 → ~70). I have proven this myself in my case beyond all reasonable doubt, but results may vary.

If you notice temperature elevation after running these scripts, you could:

- **Increase the “timeout /t 10”, I don’t know if this works, I think I tested it.**

- **Run these scripts on a Raspberry Pi Mini-Computer, with a sizable microSD card (like this one, *not an affiliate link*).**

  I myself have an entire linux PC just for running my archive scripts, which I access remotely via FileZilla, or NoMachine remote desktop.

  If you run multiple scripts, you might want to set up a FileZilla FTP server (easier than it sounds) which you can access from your main PC and transfer files from, or just upload them from there for convenience.

- **Convert the script to another code language like Python; if you don’t know how, it would probably be easy to do with AI (I will do it eventually).**

Linux

The main distinction here is that .bat files will not run, and you will need to use .sh files instead.

Also, important to note, you must create your own .sh file on linux (just save the template in a .txt file, and change the extension to .sh).
If you download a .sh file you made on windows, and send it to linux, it will have file permission errors.

Also, yt-dlp.exe CANNOT be referenced, it must say “yt-dlp” in the .sh file as given in the template below.

Linux .sh Loop Template

```
while true
do
yt-dlp --match-filter is_live --add-metadata --write-thumbnail --embed-thumbnail
--convert-thumbnail jpg --write-info-json --compat-options filename-sanitization
--download-archive done.txt --merge-output-format "mp4"     [REPLACE WITH URL OF
LIVESTREAMER, THIS CAN BE YOUTUBE, TWITCH, ETC.]
  sleep 20
done
```

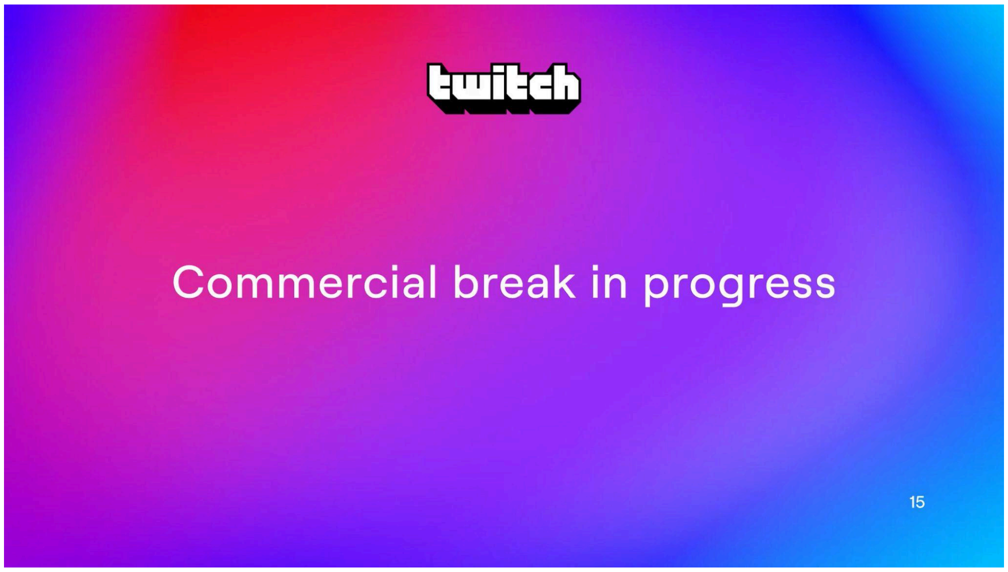
Save this as a text file, and rename the file extension to .sh.

The Problem with Twitch Vods

Twitch automatically embeds ads into most streams, and while you *can* use yt-dlp to archive Twitch streamers, it is not ideal for a few reasons:

- **Twitch vods with ads enabled will have “Commercial Break in Progress” pop ups, which last for the length of the ad, most of the time.** (Not all streamers include ads).

- **These ads are embedded into the stream vod that yt-dlp creates on your PC.**



- **If an ad happens to be at the start of the yt-dlp vod which is downloaded, it will completely ruin the audio track; causing audio glitching & lower-pitched audio for the entire vod** (which cannot be fixed by re-encoding it, but that may be a better option than leaving it raw).

Yt-dlp does not have a way to evade these embedded Twitch ads, so unless you’re willing to deal with those issues, you’ll need to use another program.

Streamlink

This is the ideal program to use with Twitch streams, specifically for stream loops. It avoids all the main issues with using yt-dlp for Twitch.

Streamlink loops will be in a .sh format, meaning you must download GitBash to operate them on Windows (see “SPECIAL TOOLS” section).

Streamlink .sh Loop Template

```
while true
do
  date=$(date '+%Y-%m-%d %H;%M;%S')

  streamlink "--force-progress" https://www.twitch.tv/[INSERT CHANNEL] 1080p60
--twitch-reexec-on-ad "-o [INSERT CHANNEL NAME] (live) {time:%Y-s%m-%d
%H%M%S} [{id}].mp4"

  echo "Tried $x times"
  x=$(( $x + 1 ))
  sleep 20
done
```

Just replace the whole Twitch URL with the streamer you choose, and add their name to the filename output format.

- **If they stream at a lower resolution than 1080p60, you can specify it like with 720p60.**

- Or you can type “best” and it will typically choose the best format (but you’d want to ensure it does, it may bug out and choose a different format nowadays).

You can also reference yt-dlp within the streamlink script, to acquire .info.json and thumbnail files for the vod. Of course, either have yt-dlp.exe in the folder, or in your PATH.

*yt-dlp --write-info-json --skip-download https://twitch.tv/https://www.twitch.tv/**[INSERT CHANNEL]** && streamlink….*

^^ The yt-dlp command comes before the streamlink one, and you must have –skip-download so it avoids using yt-dlp to download the vod.

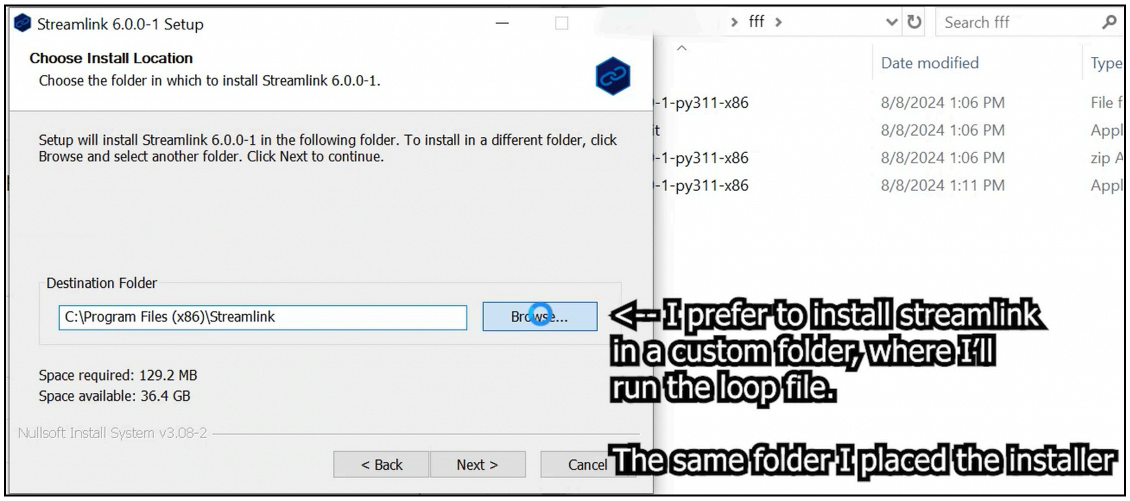
Save this as a text file, and rename the file extension to .sh. Then install streamlink.

Main Method (TTVLOL Re-exec)

For this example, I will be providing an older version of the program for compatibility, as someone who tested this wasn’t able to have it work with a newer version (your results may vary; if you try a newer version of streamlink, you have to find a newer Twitch.py file).

1. **Install Streamlink (version 6.0.0)**

   https://github.com/streamlink/windows-builds/releases/tag/6.0.0-1



Streamlink 6.0.0-1 Setup

Choose Install Location

Choose the folder in which to install Streamlink 6.0.0-1.

Setup will install Streamlink 6.0.0-1 in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

Destination Folder

C:\Program Files (x86)\Streamlink

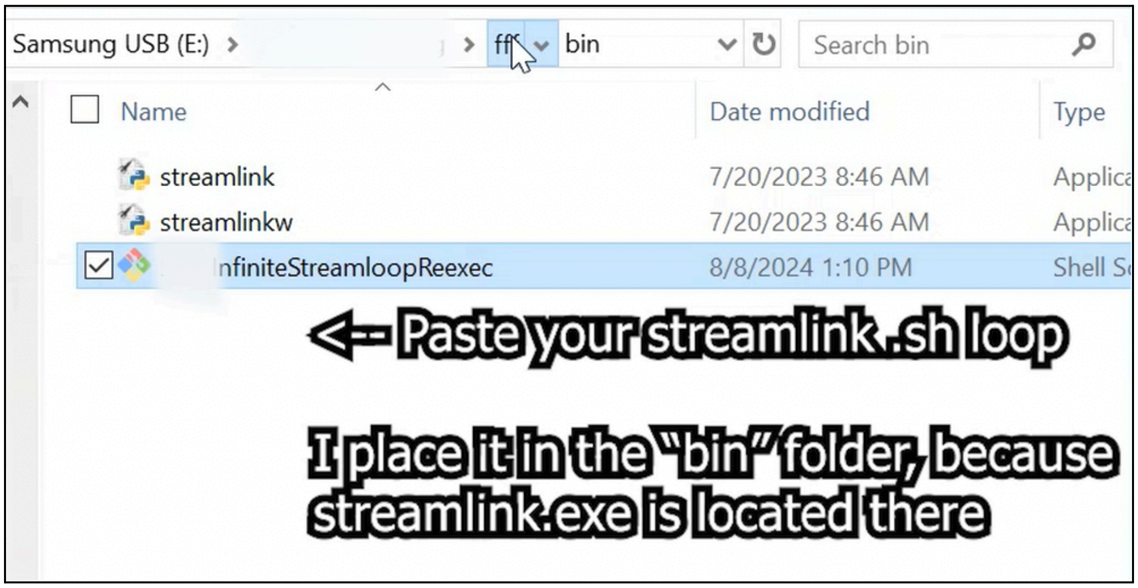
Space required: 129.2 MB
Space available: 36.4 GB

Nullsoft Install System v3.08-2

	Date modified	Type
-1-py311-x86	8/8/2024 1:06 PM	File f
t	8/8/2024 1:06 PM	Appl
-1-py311-x86	8/8/2024 1:06 PM	zip A
-1-py311-x86	8/8/2024 1:11 PM	Appl

Browse...
< Back
Next >
Cancel
fff
Search fff
<-- I prefer to install streamlink in a custom folder, where I'll run the loop file.
The same folder I placed the installer

2. **Place your streamlink looping script into the "bin" folder created by streamlink (which contains streamlink.exe).**



Name	Date modified	Type
streamlink	7/20/2023 8:46 AM	Applica
streamlinkw	7/20/2023 8:46 AM	Applica
nfiniteStreamloopReexec	8/8/2024 1:10 PM	Shell S

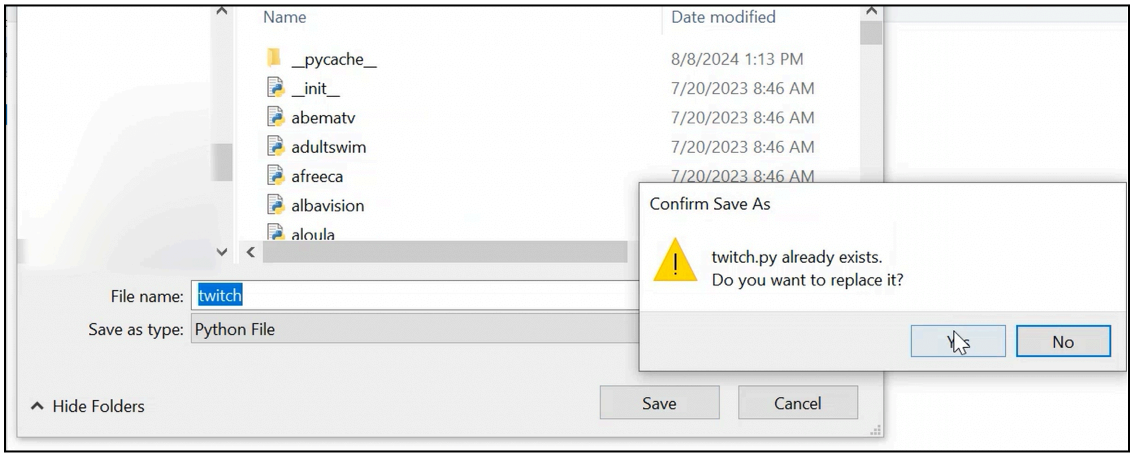
Samsung USB (E:)
fff
bin
Search bin
<-- Paste your streamlink .sh loop
I place it in the "bin" folder, because streamlink.exe is located there

3. **Download the Twitch TTVLOL replacement for twitch.py.**
I have included the exact file for streamlink version 6.0.0:

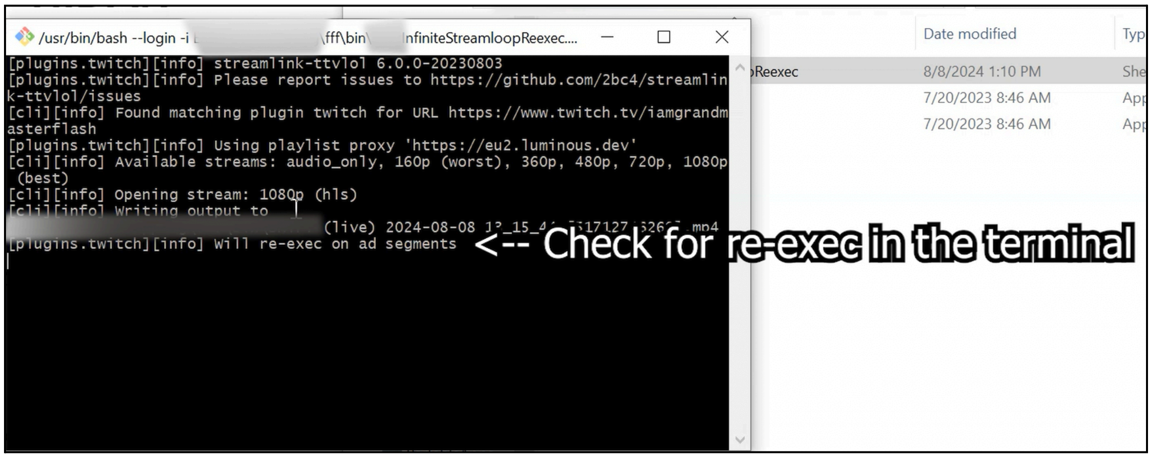
https://drive.google.com/file/d/1CfKVNLjlpMoWzKdPkYJ5BlpNEuZww35B

4. **Place the new twitch.py into this directory, and replace the current one:**

[[*Folder Directory]]/pkgs/streamlink/plugins/twitch.py*



5. **Run your looping script from the "bin" directory.** Probably create a shortcut to open it elsewhere.


/usr/bin/bash --login -i [redacted]\fff\bin\[redacted]nfiniteStreamloopReexec....
Date modified
Typ
oReexec
8/8/2024 1:10 PM
She
7/20/2023 8:46 AM
App
7/20/2023 8:46 AM
App
<-- Check for re-exec in the terminal

```
[plugins.twitch][info] streamlink-ttvlol 6.0.0-20230803
[plugins.twitch][info] Please report issues to https://github.com/2bc4/streamlin
k-ttvlol/issues
[cli][info] Found matching plugin twitch for URL https://www.twitch.tv/iamgrandm
asterflash
[plugins.twitch][info] Using playlist proxy 'https://eu2.luminous.dev'
[cli][info] Available streams: audio_only, 160p (worst), 360p, 480p, 720p, 1080p
 (best)
[cli][info] Opening stream: 1080p (hls)
[cli][info] Writing output to
[redacted] (live) 2024-08-08 13_15_4[illegible]_[illegible]17127[illegible]3262[illegible].mp4
[plugins.twitch][info] Will re-exec on ad segments
```

<u>Here is a start to finish tutorial to set streamlink up. 2 Minute long video</u>

🎬 Streamlink TTVLOL Tutorial.mp4

If you still get ads, even with a newer version (if you want to try that option), you will have to use the Oauth token method below.

<u>Twitch Turbo OAuth Method (most reliable)</u>

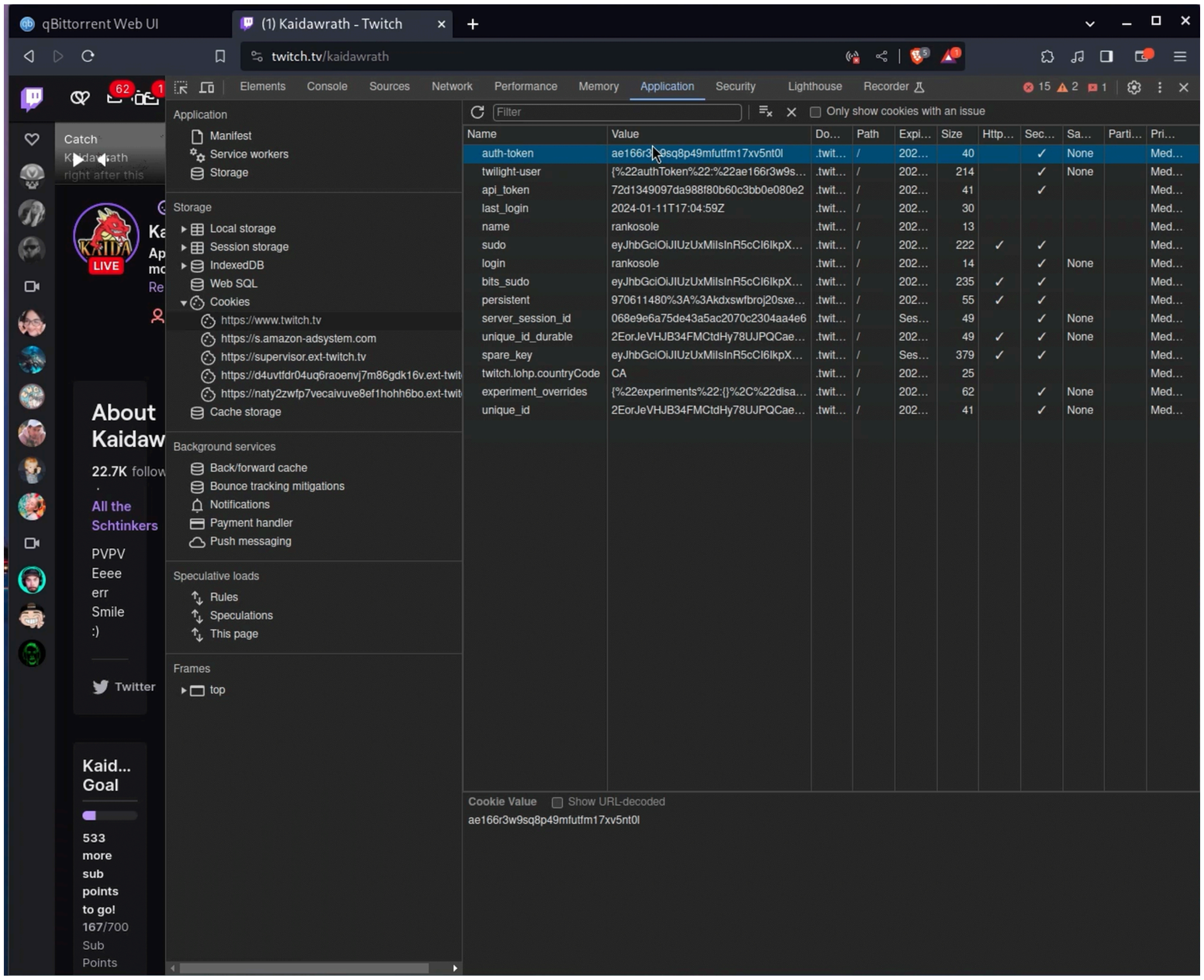
If you happen to pay for Twitch turbo, or *know someone that does*, this will guarantee you can download any twitch streams without ads.

It's non-intrusive to acquire these tokens, and nothing can be done with them besides verifying your identity to Twitch's API, so <u>I usually ask other people for Oauth tokens</u>, and when they expire I can just ask them to get a new one for me.

All you have to do is be logged into the Twitch account with Twitch Turbo (on PC, for this guide it will be in Chrome/Brave browsers), and find it by inspecting element.

Then just copy the auth-token line within the cookies. Nothing else.

🎬 **Oauth Acquisition.mp4**



Name	Value	Do…	Path	Expi…	Size	Http…	Sec…	Sa…	Parti…	Pri…
auth-token	ae166r3w9sq8p49mfutfm17xv5nt0l	.twit…	/	202…	40		✓	None		Med…
twilight-user	{%22authToken%22:%22ae166r3w9s…	.twit…	/	202…	214		✓	None		Med…
api_token	72d1349097da988f80b60c3bb0e080e2	.twit…	/	202…	41		✓			Med…
last_login	2024-01-11T17:04:59Z	.twit…	/	202…	30					Med…
name	rankosole	.twit…	/	202…	13					Med…
sudo	eyJhbGciOiJIUzUxMiIsInR5cCI6IkpX…	.twit…	/	202…	222	✓	✓			Med…
login	rankosole	.twit…	/	202…	14		✓	None		Med…
bits_sudo	eyJhbGciOiJIUzUxMiIsInR5cCI6IkpX…	.twit…	/	202…	235	✓	✓			Med…
persistent	970611480%3A%3Akdxswfbroj20sxe…	.twit…	/	202…	55	✓	✓			Med…
server_session_id	068e9e6a75de43a5ac2070c2304aa4e6	.twit…	/	Ses…	49		✓	None		Med…
unique_id_durable	2EorJeVHJB34FMCtdHy78UJPQCae…	.twit…	/	202…	49	✓	✓	None		Med…
spare_key	eyJhbGciOiJIUzUxMiIsInR5cCI6IkpX…	.twit…	/	Ses…	379	✓	✓			Med…
twitch.lohp.countryCode	CA	.twit…	/	202…	25					Med…
experiment_overrides	{%22experiments%22:{}%2C%22disa…	.twit…	/	202…	62		✓	None		Med…
unique_id	2EorJeVHJB34FMCtdHy78UJPQCae…	.twit…	/	202…	41		✓	None		Med…

Cookie Value

ae166r3w9sq8p49mfutfm17xv5nt0l

The person who obtained the Oauth must <u>NOT</u> log out of their account where they obtained it (like a browser), otherwise the Oauth will become invalid, and they'll have to find another one.

An Oauth might look like this, this is a scrambled example:

r4mkfayczgv9qs2aizo1e31nhvw760

To use this, add this argument to your streamlink .sh loop

"--twitch-api-header=Authorization=OAuth [INSERT OAUTH TOKEN]"

Streamlink Oauth .sh Loop Template

---

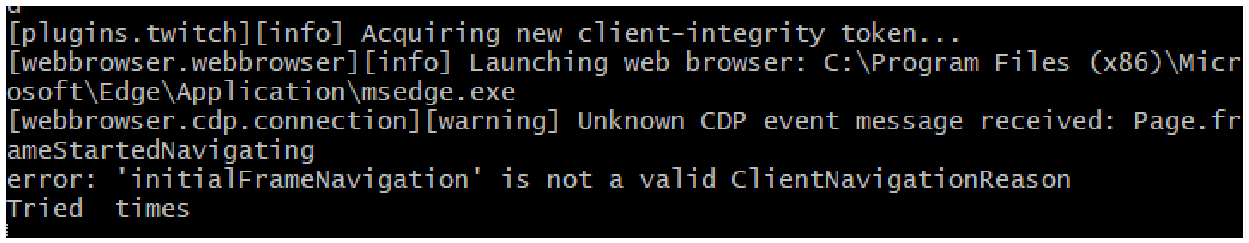
while true
do
  date=$(date '+%Y-%m-%d %H;%M;%S')

  streamlink "--force-progress" "--twitch-api-header=Authorization=OAuth ***[INSERT OAUTH TOKEN]****"* *[https://www.twitch.tv/**[INSERT CHANNEL]**](https://www.twitch.tv/)* *1080p60 "-o* ***[INSERT CHANNEL NAME]*** *(live) {time:%Y-s%m-%d %H%M%S} [{id}].mp4"*

  echo "Tried $x times"
  x=$(($x + 1))
  sleep 20
done

If the Oauth expires, either because Twitch Turbo was cancelled/not paid for, or most notably, the person logged out of their twitch account where the Oauth was obtained, you will have to find another one, but it only takes a minute to find.

This is what the error looks like: it's not readily apparent this is what it means.



```
[plugins.twitch][info] Acquiring new client-integrity token...
[webbrowser.webbrowser][info] Launching web browser: C:\Program Files (x86)\Micr
osoft\Edge\Application\msedge.exe
[webbrowser.cdp.connection][warning] Unknown CDP event message received: Page.fr
ameStartedNavigating
error: 'initialFrameNavigation' is not a valid ClientNavigationReason
Tried  times
```

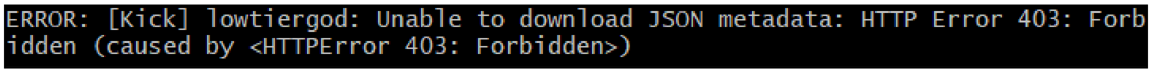
Kick Streams

Kick streams can have more issues when downloaded with yt-dlp stream loops, but they are manageable.

This is not universally the case, as my friend has this work completely fine. I used to be unable to get it to work on linux, despite not using a VPN. But all you have to do is install curl-cffi.

<u>In my case, it would:</u>

- **Will typically give this error:**



```
ERROR: [Kick] lowtiergod: Unable to download JSON metadata: HTTP Error 403: Forb
idden (caused by <HTTPError 403: Forbidden>)
```

While this has been patched by yt-dlp before, I've noticed this error acutely, and they may have a mechanism to block certain IPs, because connecting to NordVPN actually fixed my issue in that case (adding cookies with a Kick account logged in did not work). But other times, the VPN makes no difference (such as on my linux PC).

- **Will split the entire stream into hundreds/thousands of files which cannot be joined practically, making this entirely useless.**

[To Fix the 403 Forbidden Errors:]

On linux, just paste this command:

pip install curl-cffi

This immediately fixed the 403 forbidden error on my end.

You can also try these other options if needed:

- Update yt-dlp. This fixed it for me at least one time
- **--cookies-from-browser firefox (logged into a Kick account)**
- **--user-agent "[USERAGENTSTRING]"**
  This can be used in conjunction with your --cookies-from-browser command to include more info to pass checks.
  Find your UserAgentString:
  https://whatismybrowser.com/detect/what-is-my-user-agent/

To deal with the split stream issues of yt-dlp (which you may want to avoid the risk of entirely), I've had a new script coded in which yt-dlp can 'pipe' the m3u8 url (from the kick stream) to send into streamlink. Streamlink will then download the vod with the format command it's given.
The script loops at a set interval of your choosing.

Streamlink is **not** coded to work with kick.com directly (rather it finds the m3u8 url on the *backend* of kick and then uses that to download the stream), so this uses the best functions of both programs for this.

Kick Python-yt-dlp-Streamlink Loop Template

- This script requires a python shell (like IDLE), yt-dlp, and streamlink from previous steps.
- You **must** have yt-dlp.exe and streamlink.exe in your PATH.
  If you haven't set them up, see the "*Set yt-dlp/ffmpeg to be Accessible Anywhere*" section at the beginning. It's always best to have yt-dlp, ffmpeg, and streamlink all listed in your environmental variables.

You can also download Kick vods after the fact, with the links to the vods, just like any other videos.

RENDER YT-DLP JSONS AS A FAKE YOUTUBE PAGE (v4.17.8)

This is the most unique, and complex, script I've generated so far.
The best feature is being able to sort comments by Most Likes, Most Replies, and Longest Length, and fully-simulate a youtube video page, with embedded audio and thumbnails **all in a standalone file**.

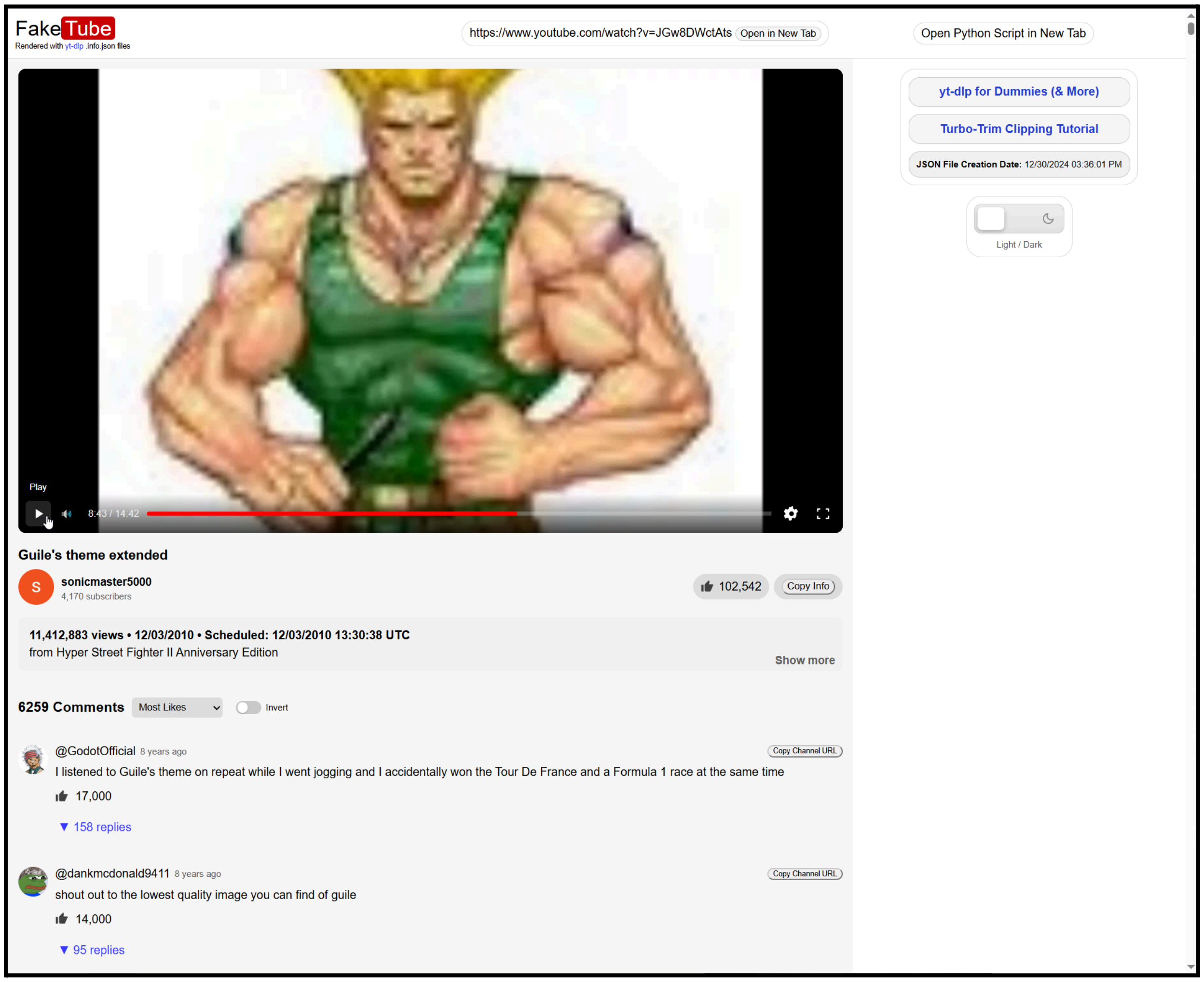
It allows you to actually visualize what the .info.json files store from downloaded Youtube videos. Without needing the video itself, or anything else for that matter (once the HTML file is created). Though I highly recommend including thumbnails, and channel uploader images, in the same directory you're running this script (all jsons, all thumbnails, and a single channel pfp metadata file in one folder).

Most useful for archiving deleted videos/terminated channels that you proactively backed up. ***For a more detailed guide, make sure to read the comment block at the top of the script.***

Python Script Download

Render YouTube Comments as an HTML Webpage v4.17.8.py

Since Version 3.8.8, a bunch of UI and User Experience improvements have been made. Audio and thumbnails and uploader images can be embedded, and a dark mode toggle was added. This took much longer than you'd expect; AI is not fun to troubleshoot with when you don't know any web developers.



FakeTube
Rendered with yt-dlp .info json files

https://www.youtube.com/watch?v=JGw8DWctAts Open in New Tab

Open Python Script in New Tab

yt-dlp for Dummies (& More)

Turbo-Trim Clipping Tutorial

JSON File Creation Date: 12/30/2024 03:36:01 PM

Light / Dark

Play

8:43 / 14:42

Guile's theme extended

sonicmaster5000
4,170 subscribers

102,542 Copy Info

11,412,883 views • 12/03/2010 • Scheduled: 12/03/2010 13:30:38 UTC
from Hyper Street Fighter II Anniversary Edition

Show more

6259 Comments Most Likes Invert

@GodotOfficial 8 years ago Copy Channel URL
I listened to Guile's theme on repeat while I went jogging and I accidentally won the Tour De France and a Formula 1 race at the same time
17,000
▼ 158 replies

@dankmcdonald9411 8 years ago Copy Channel URL
shout out to the lowest quality image you can find of guile
14,000
▼ 95 replies

The script takes a directory full of yt-dlp JSON files (ideally with comments written) and makes HTML versions out of most of the unique data contained inside.

Audio and thumbnails, when containing the yt-dlp formatting for filenames, will be embedded as well if provided in the same directory for each .info.json (using base64).

It Features:

- **A fully-functional comment section, with some unique features not available on YouTube.** This is not replicated by the closest alternative, the Wayback Machine. And Raw yt-dlp jsons may contain comments, but are not enough.

- **4 Unique ways to sort comments (Chronological order/'Top' not included, dependent on info only youtube owns), as well as an "Invert" button to flip results for large files.**

- **Various buttons to copy uploader/video info, download embedded audio/thumbnails, download profile pictures, etc.**

- **A fully-functional audio player, quite similar to Youtube with similar toolbars and keyboard shortcuts. Accomplished via base64 HTML embedding.** Audio can be boosted by right-clicking the mute button.

- **Various links to open relevant Youtube links (video url, channel url, commenter URL and/or channel handle).**

- **A dark mode toggle, and links to this document, a related one and the script itself on Pastebin.**

Three Minor Limitations

1. **Comments AND replies cannot be sorted chronologically.** Unfortunately .info.jsons files don't store comment timestamps accurately for some reason – likely Youtube's proprietary system, so they don't convert to UTC time correctly to be sorted. Sometimes they convert to the correct date on the comments, but this is inconsistent between jsons, and no matter what the UTC time is always 00:00:00. So I scrapped showing timestamps entirely for comments.

2. **Uploader profile pictures are not guaranteed to display,** because the links are not stored in the JSON files unless the uploader made a comment/reply on their own video (this is why commenters have profile pictures, and you can open them in a new tab).

3. **Video files cannot be embedded into the HTML.**

Previously, thumbnails/audio wasn't embedded, nor uploader profile pics, but base64 has made this possible and very easy to accomplish at the scale of hundreds of thousands, when you use the appropriate yt-dlp arguments.

Video files wouldn't work due to base64 limitations. Even if it did, only smaller videos would work well. And having HTML files in hundreds of mb would get messy, playback slowly or break, and be impractical to archive at a grand scale.
If the video still exists on youtube, you can just click on the open in new tab button, or manually play the video file often included alongside the HTML for archival.

I can think of two best use-cases for this

- You want to get more info on individual videos, such as unique comment sorting (*Most Likes*, *Most Replies*, *Longest Length*, *Alphabetically*) or tags/categories and other info.

  Or you just like the idea of viewing a fake youtube HTML page.
  (You can get the same info retrieved in mass across hundreds of videos with this script, but it won't visualize, and it doesn't save/sort comments).

- You proactively archive channels which may get terminated in the future, and want to simulate what the video looks like if/when that ever happens. **So you can archive a video, with an HTML file, for the video to simulate its main elements, asides from the actual video playback (which is a minor downside).**

  This is objectively superior to The Wayback Machine for two reasons:

  1. Wayback Machine doesn't archive comments correctly (ever), and sorting wouldn't work anyway if it did. Although, this script can't sort by top comment/newest; this effectively isn't possible anyway with Youtube's system.
  2. Not *every single video* gets a snapshot stored on the site. You can create json files for every video on a youtube channel without even downloading the videos, and simulate the comments.

I archive some controversial Youtube channels for preservation, so if they get terminated, I can simulate what the page looked like and view the comments, and potentially play the audio on its own.

The same goes for video game music channels which are often maliciously flagged by Nintendo, which need to be preserved at scale and mirrored, but I digress.

I'll start uploading HTML files with all my Internet Archive posts, as well as the jsons and thumbnails; that's one potential use I recommend. Google Drive (free 15gb) is decent for huge amounts of HTML files, but only if audio isn't embedded, then Internet Archive is best.

<u>If you *just* want metadata files on an entire channel's videos, run this command.</u>

This is how you can make these WITHOUT downloading an entire channel's actual videos.

```
yt-dlp --skip-download --write-info-json --write-comments
<<CHANNEL/PLAYLIST URL>>
```

- If you only want videos, not shorts or live streams from a channel, click into their videos tab and then copy the URL like this: https://www.youtube.com/@channelname/videos

- It should also include a file with the channel name as the title, that has public channel metadata like subscribers and the about section. It isn't relevant to this script but has some unique info.

RENDER YT-DLP LIVE-CHAT JSONS AS A FAKE CHAT PAGE

This is a lesser-known feature of YT-DLP. You can archive a chat for any livestream, which itself can be automated within a looping .bat or .sh that runs on a youtube channel's live stream tab (to auto-archive all livestream chats).

Here is a live_chat .bat file which runs on Windows, you can insert the handle of the youtuber who you want to automatically archive the stream chats of.

It will automatically save the live chats, which you can use later.

This script, much like the other "RENDER YT-DLP JSONS" script, will render all of the useful information stored inside the .livechat.json file into a functional representation of the stream chat.

Download

Render YouTube Livechat as an HTML Webpage MkII.py

There are some limitations

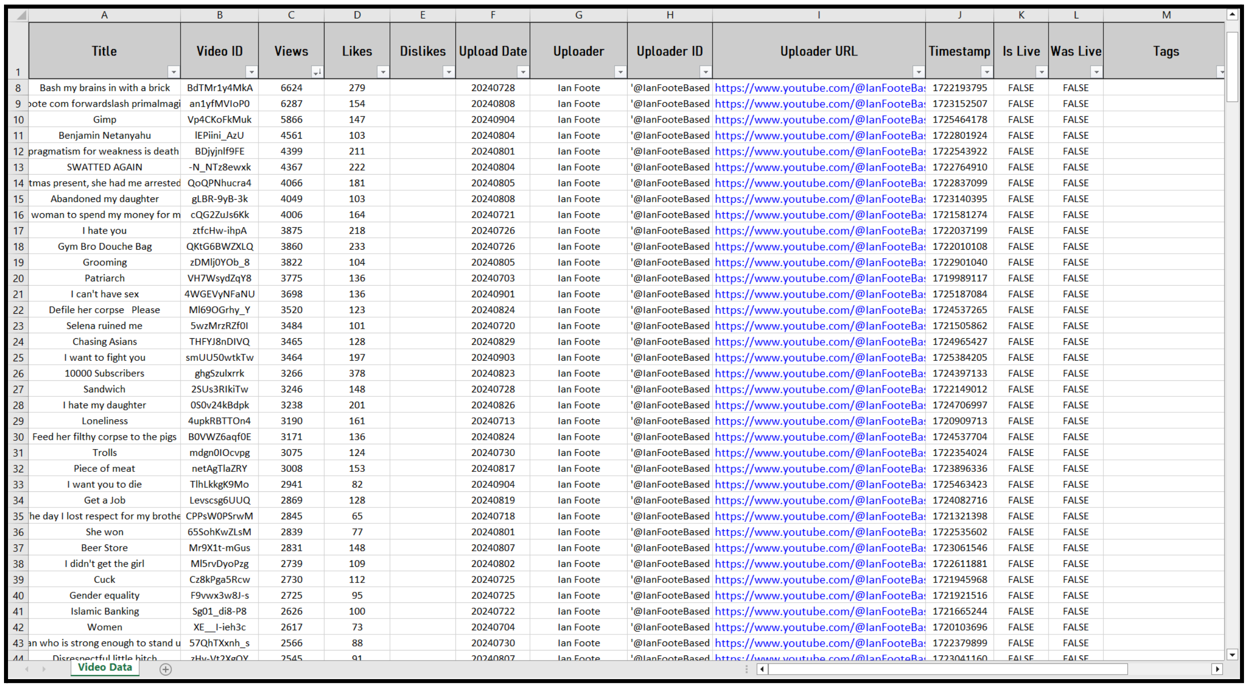
1. **Channel profile pictures are dependent on the original person not changing their profile picture, or deleting/losing their account.** Old HTML live chats will likely contain several "N/A" alt texts as placeholders.
   There is no way to make this work, besides screen recording, since livechat files don't create their own thumbnail files stored within the json. It relies on the original google URL.

2. **Channel profiles cannot be opened in a new tab, like with the .info.json HTML script.** This is simply due to .live_chat.json files not saving uploader handles or channel IDs of chat users.

You'd have to backtrace their channel name to find the original on youtube, which is quite difficult given Youtube's shoddy search engine.

3. **There are only a few data points per live_chat comment:**
   Thumbnail URL, author name (no links), timestamp (relative to the stream, not a universal time), and message text.

INSTANT METADATA VISUALIZATION HACK



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Title	Video ID	Views	Likes	Dislikes	Upload Date	Uploader	Uploader ID	Uploader URL	Timestamp	Is Live	Was Live	Tags
8	Bash my brains in with a brick	BdTMr1y4MkA	6624	279		20240728	Ian Foote	'@IanFooteBased	https://www.youtube.com/@IanFooteBa	1722193795	FALSE	FALSE	
9	ɔote com forwardslash primalmagi	an1yfMVIoP0	6287	154		20240808	Ian Foote	'@IanFooteBased	https://www.youtube.com/@IanFooteBa	1723152507	FALSE	FALSE	
10	Gimp	Vp4CKoFkMuk	5866	147		20240904	Ian Foote	'@IanFooteBased	https://www.youtube.com/@IanFooteBa	1725464178	FALSE	FALSE	
11	Benjamin Netanyahu	lEPiini_AzU	4561	103		20240804	Ian Foote	'@IanFooteBased	https://www.youtube.com/@IanFooteBa	1722801924	FALSE	FALSE	
12	pragmatism for weakness is death	BDjyjnlf9FE	4399	211		20240801	Ian Foote	'@IanFooteBased	https://www.youtube.com/@IanFooteBa	1722543922	FALSE	FALSE	
13	SWATTED AGAIN	-N_NTz8ewxk	4367	222		20240804	Ian Foote	'@IanFooteBased	https://www.youtube.com/@IanFooteBa	1722764910	FALSE	FALSE	
14	tmas present, she had me arrested	QoQPNhucra4	4066	181		20240805	Ian Foote	'@IanFooteBased	https://www.youtube.com/@IanFooteBa	1722837099	FALSE	FALSE	
15	Abandoned my daughter	gLBR-9yB-3k	4049	103		20240808	Ian Foote	'@IanFooteBased	https://www.youtube.com/@IanFooteBa	1723140395	FALSE	FALSE	
16	woman to spend my money for m	cQG2ZuJs6Kk	4006	164		20240721	Ian Foote	'@IanFooteBased	https://www.youtube.com/@IanFooteBa	1721581274	FALSE	FALSE	
17	I hate you	ztfcHw-ihpA	3875	218		20240726	Ian Foote	'@IanFooteBased	https://www.youtube.com/@IanFooteBa	1722037199	FALSE	FALSE	
18	Gym Bro Douche Bag	QKtG6BWZXLQ	3860	233		20240726	Ian Foote	'@IanFooteBased	https://www.youtube.com/@IanFooteBa	1722010108	FALSE	FALSE	
19	Grooming	zDMlj0YOb_8	3822	104		20240805	Ian Foote	'@IanFooteBased	https://www.youtube.com/@IanFooteBa	1722901040	FALSE	FALSE	
20	Patriarch	VH7WsydZqY8	3775	136		20240703	Ian Foote	'@IanFooteBased	https://www.youtube.com/@IanFooteBa	1719989117	FALSE	FALSE	
21	I can't have sex	4WGEVyNFaNU	3698	136		20240901	Ian Foote	'@IanFooteBased	https://www.youtube.com/@IanFooteBa	1725187084	FALSE	FALSE	
22	Defile her corpse Please	Ml69OGrhy_Y	3520	123		20240824	Ian Foote	'@IanFooteBased	https://www.youtube.com/@IanFooteBa	1724537265	FALSE	FALSE	
23	Selena ruined me	5wzMrzRZf0I	3484	101		20240720	Ian Foote	'@IanFooteBased	https://www.youtube.com/@IanFooteBa	1721505862	FALSE	FALSE	
24	Chasing Asians	THFYJ8nDIVQ	3465	128		20240829	Ian Foote	'@IanFooteBased	https://www.youtube.com/@IanFooteBa	1724965427	FALSE	FALSE	
25	I want to fight you	smUU50wtkTw	3464	197		20240903	Ian Foote	'@IanFooteBased	https://www.youtube.com/@IanFooteBa	1725384205	FALSE	FALSE	
26	10000 Subscribers	ghgSzulxrrk	3266	378		20240823	Ian Foote	'@IanFooteBased	https://www.youtube.com/@IanFooteBa	1724397133	FALSE	FALSE	
27	Sandwich	2SUs3RIkiTw	3246	148		20240728	Ian Foote	'@IanFooteBased	https://www.youtube.com/@IanFooteBa	1722149012	FALSE	FALSE	
28	I hate my daughter	0S0v24kBdpk	3238	201		20240826	Ian Foote	'@IanFooteBased	https://www.youtube.com/@IanFooteBa	1724706997	FALSE	FALSE	
29	Loneliness	4upkRBTTOn4	3190	161		20240713	Ian Foote	'@IanFooteBased	https://www.youtube.com/@IanFooteBa	1720909713	FALSE	FALSE	
30	Feed her filthy corpse to the pigs	B0VWZ6aqf0E	3171	136		20240824	Ian Foote	'@IanFooteBased	https://www.youtube.com/@IanFooteBa	1724537704	FALSE	FALSE	
31	Trolls	mdgn0IOcvpg	3075	124		20240730	Ian Foote	'@IanFooteBased	https://www.youtube.com/@IanFooteBa	1722354024	FALSE	FALSE	
32	Piece of meat	netAgTlaZRY	3008	153		20240817	Ian Foote	'@IanFooteBased	https://www.youtube.com/@IanFooteBa	1723896336	FALSE	FALSE	
33	I want you to die	TlhLkkgK9Mo	2941	82		20240904	Ian Foote	'@IanFooteBased	https://www.youtube.com/@IanFooteBa	1725463423	FALSE	FALSE	
34	Get a Job	Levscsg6UUQ	2869	128		20240819	Ian Foote	'@IanFooteBased	https://www.youtube.com/@IanFooteBa	1724082716	FALSE	FALSE	
35	he day I lost respect for my brothe	CPPsW0PSrwM	2845	65		20240718	Ian Foote	'@IanFooteBased	https://www.youtube.com/@IanFooteBa	1721321398	FALSE	FALSE	
36	She won	65SohKwZLsM	2839	77		20240801	Ian Foote	'@IanFooteBased	https://www.youtube.com/@IanFooteBa	1722535602	FALSE	FALSE	
37	Beer Store	Mr9X1t-mGus	2831	148		20240807	Ian Foote	'@IanFooteBased	https://www.youtube.com/@IanFooteBa	1723061546	FALSE	FALSE	
38	I didn't get the girl	Ml5rvDyoPzg	2739	109		20240802	Ian Foote	'@IanFooteBased	https://www.youtube.com/@IanFooteBa	1722611881	FALSE	FALSE	
39	Cuck	Cz8kPga5Rcw	2730	112		20240725	Ian Foote	'@IanFooteBased	https://www.youtube.com/@IanFooteBa	1721945968	FALSE	FALSE	
40	Gender equality	F9vwx3w8J-s	2725	95		20240725	Ian Foote	'@IanFooteBased	https://www.youtube.com/@IanFooteBa	1721921516	FALSE	FALSE	
41	Islamic Banking	Sg01_di8-P8	2626	100		20240722	Ian Foote	'@IanFooteBased	https://www.youtube.com/@IanFooteBa	1721665244	FALSE	FALSE	
42	Women	XE__I-ieh3c	2617	73		20240704	Ian Foote	'@IanFooteBased	https://www.youtube.com/@IanFooteBa	1720103696	FALSE	FALSE	
43	an who is strong enough to stand u	57QhTXxnh_s	2566	88		20240730	Ian Foote	'@IanFooteBased	https://www.youtube.com/@IanFooteBa	1722379899	FALSE	FALSE	
44	Disrespectful little bitch	zHy-Vt2XgOY	2545	91		20240807	Ian Foote	'@IanFooteBased	https://www.youtube.com/@IanFooteBa	1723041160	FALSE	FALSE	

Video Data

(if you use the script, you'd probably want to delete the dislike column, unless you have old pre-2022 json files)

yt-dlp has the ability to export all public metadata from any youtube video. Views, likes, upload date, tags, description and more.

This can be done using the “--write-info-json” command. I would recommend using “--write-comments” too, it saves all of them. *Very few archivists use this, and they miss the chance to save comments of a (later on) terminated Youtube channel.*

If you *just* want metadata files on an entire channel’s videos, run this command.

This is how you can make these WITHOUT downloading an entire channel’s actual videos.

```
yt-dlp --skip-download --write-info-json --write-comments
<<CHANNEL/PLAYLIST URL>>
```

- If you only want videos, without including shorts or live streams from a channel, click into their videos tab and then copy the URL like this: https://www.youtube.com/@channelname/videos

- It should also include a file with the channel name as the title, that has public channel metadata like subscribers and the about section. It isn’t relevant to this script but has some unique info.

Designed with GPT-4o, this script takes all the json metadata files for a channel, and converts them into a convenient Excel spreadsheet.
If you don’t have excel, it's not hard to get. I have a guide in the script, and later on.

If you’re on linux, here’s an alternate script for .ods files [insert here]

Using this script you can see and organize the most popular / liked videos, when it was posted, tags, descriptions and more.

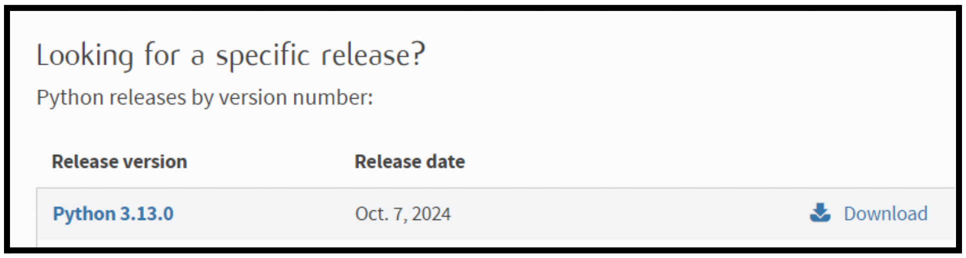
Download

Extract All Metadata from Json Files in Directory to Spreadsheet.py

Requirements to Use (just follow the steps below):

- Folder with all the .json files you want to run on.
- Python IDLE to run the program.
- GitBash to install “pandas” and “openpyxl” packages, only takes a single command to do once gitbash is installed. Really simple.
- Excel, Libreoffice, or OnlyOffice to open .xlsx files. I highly recommend just running an Office activation script, it only takes a minute. *Or if you’re on linux, you can use OpenOffice or Libreoffice.*

1. **Install Python IDLE**. With it you’ll be able to run .py files.

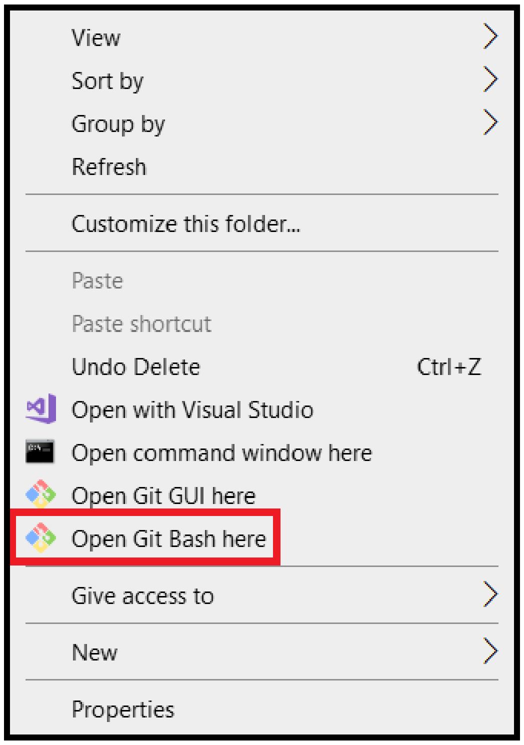


Looking for a specific release?

Python releases by version number:

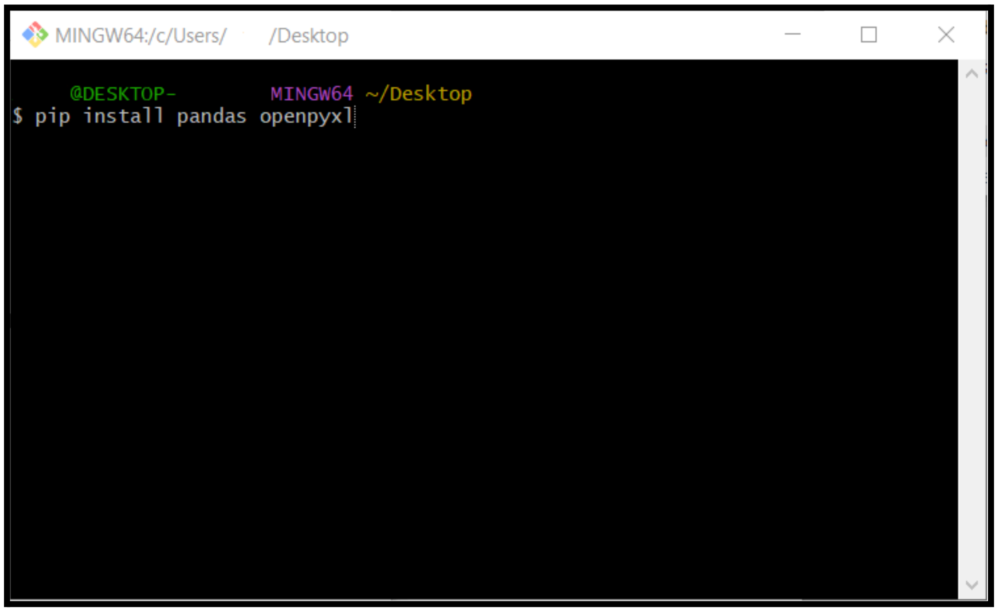
Release version	Release date	
Python 3.13.0	Oct. 7, 2024	Download

2. **Install GitBash** (unless you’re on Linux). Then open the terminal by right clicking in any folder and clicking “Open Git Bash here.”



3. **Run this command** and hit enter to install the necessary packages to use the script. Without these it can't function.

   pip install pandas openpyxl

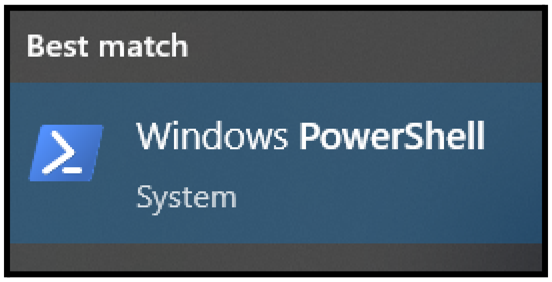


```
     @DESKTOP-        MINGW64 ~/Desktop
$ pip install pandas openpyxl
```

4. **Install (preferably) Microsoft Office.** You can use these activation scripts. Don't take my word for it, they're trusted widely on forums like reddit.

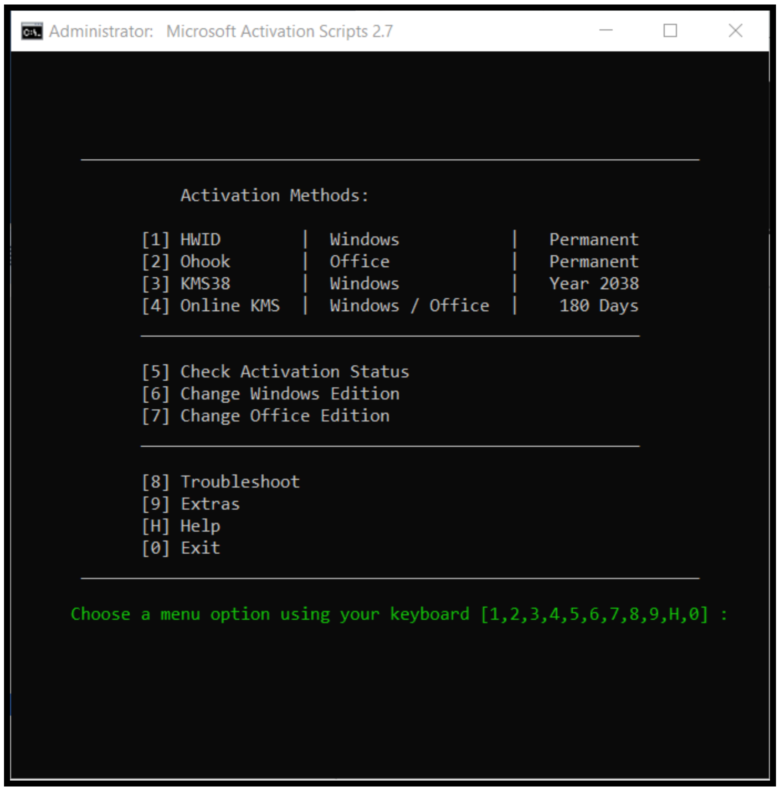
   - Setup the base Office program. You'll get Word, Excel, Publisher, Access, Outlook, and OneNote. https://www.microsoft.com/en-us/microsoft-365/download-office#download

   - Open Windows Powershell



   - Paste this command and hit enter:
     irm https://get.activated.win | iex

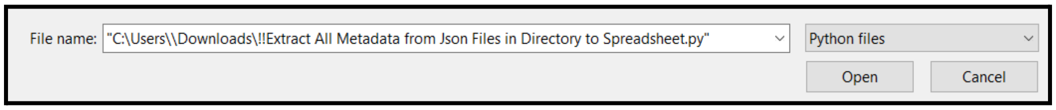
   - Type the number 2 in the new window to activate office, let it run and you're done, excel will work.



```
Administrator:  Microsoft Activation Scripts 2.7

       ______________________________________________________________

                 Activation Methods:

             [1] HWID         |  Windows          |  Permanent
             [2] Ohook        |  Office           |  Permanent
             [3] KMS38        |  Windows          |  Year 2038
             [4] Online KMS   |  Windows / Office |   180 Days
             __________________________________________________

             [5] Check Activation Status
             [6] Change Windows Edition
             [7] Change Office Edition
             __________________________________________________

             [8] Troubleshoot
             [9] Extras
             [H] Help
             [0] Exit
       ______________________________________________________________

      Choose a menu option using your keyboard [1,2,3,4,5,6,7,8,9,H,0] :
```

5. **Download the script** from here.
   https://drive.google.com/file/d/1gUUjWWYNIucIspHn_htgGRJ93u_qrked/view?usp=sharing

   - Open it either by opening the Python IDLE shell and pressing CTRL-O, then CTRL-C on the script in the folder (same as copying the file), and pressing CTRL-V inside the File name box.

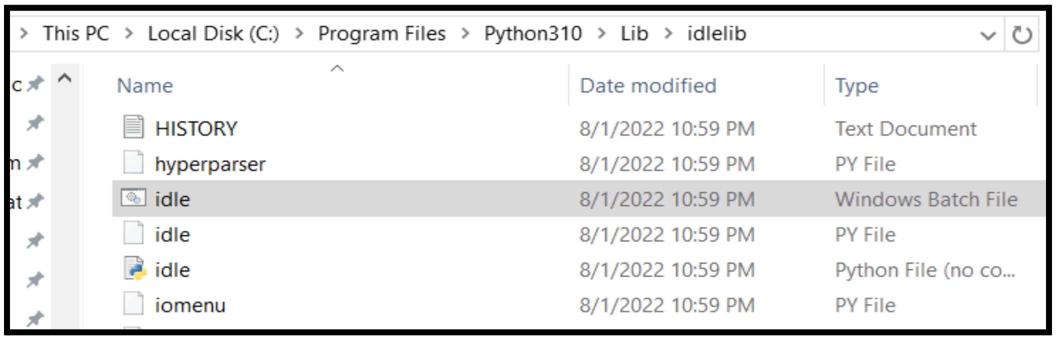


File name: "C:\Users\\Downloads\!!Extract All Metadata from Json Files in Directory to Spreadsheet.py"

   - Or by setting IDLE.exe as the default program to open .py files. This is more complicated than it should be; setting python.exe or pythonw.exe as default will not open the script in IDLE if the file is double-clicked, at least in my experience.

     You'll have to find the install directory of python, it should be inside a similar folder to this:
     C:\Program Files\Python310\Lib\idlelib

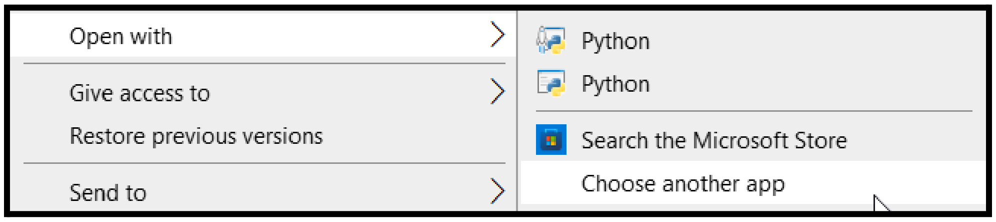
Find **idle.bat**.

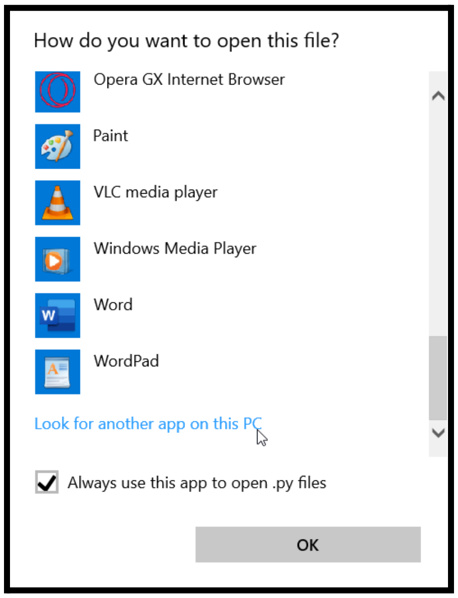


This PC > Local Disk (C:) > Program Files > Python310 > Lib > idlelib

Name	Date modified	Type
HISTORY	8/1/2022 10:59 PM	Text Document
hyperparser	8/1/2022 10:59 PM	PY File
idle	8/1/2022 10:59 PM	Windows Batch File
idle	8/1/2022 10:59 PM	PY File
idle	8/1/2022 10:59 PM	Python File (no co...
iomenu	8/1/2022 10:59 PM	PY File

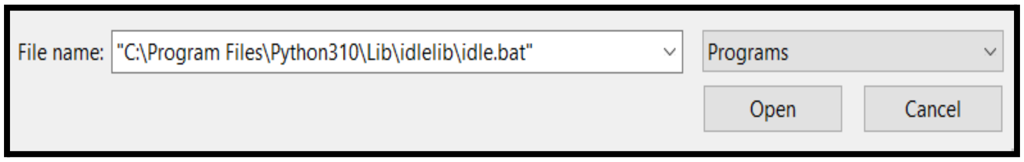
- Right click the script and Choose another app, click “More Apps” and scroll until “Look for another app on this PC”. Make sure the box is checked, then click and paste the location of idle.bat into the File name box, using the same command as copying a file.





How do you want to open this file?

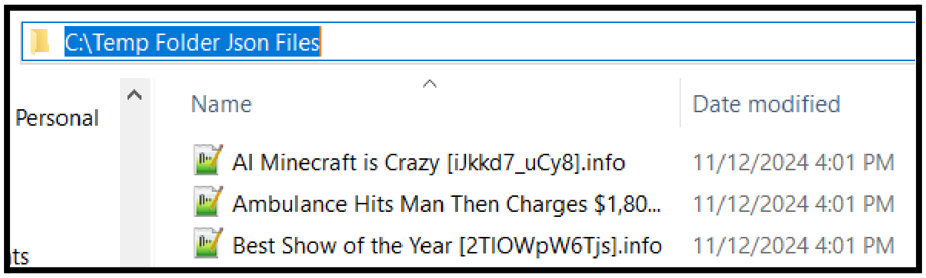
[✓] Always use this app to open .py files



File name: "C:\Program Files\Python310\Lib\idlelib\idle.bat"

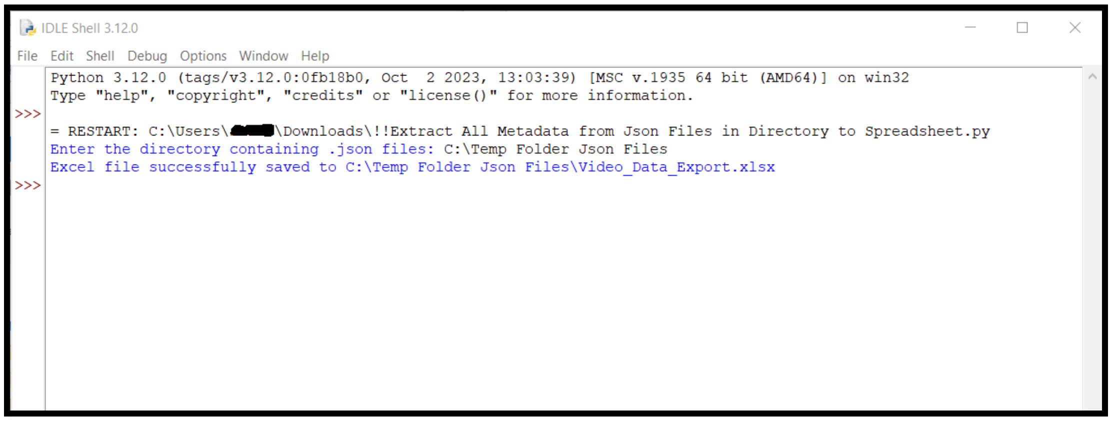
Programs

6. Open the script in IDLE just by double-clicking, press F5 on the keyboard or select “Run Module” from the Run option on the toolbar. Then copy the directory you have all of the .info.json files downloaded with yt-dlp.



C:\Temp Folder Json Files

Name	Date modified
AI Minecraft is Crazy [iJkkd7_uCy8].info	11/12/2024 4:01 PM
Ambulance Hits Man Then Charges $1,80...	11/12/2024 4:01 PM
Best Show of the Year [2TlOWpW6Tjs].info	11/12/2024 4:01 PM



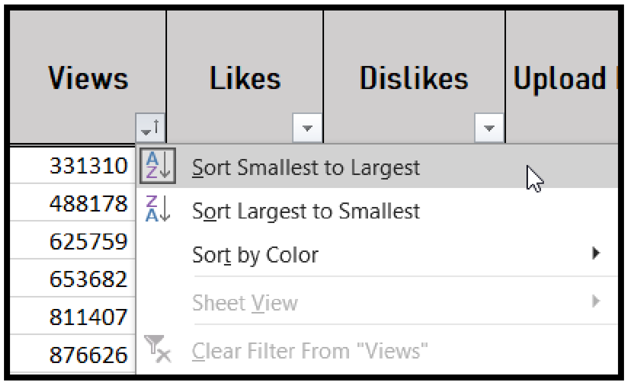
```
Python 3.12.0 (tags/v3.12.0:0fb18b0, Oct  2 2023, 13:03:39) [MSC v.1935 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\[redacted]\Downloads\!!Extract All Metadata from Json Files in Directory to Spreadsheet.py
Enter the directory containing .json files: C:\Temp Folder Json Files
Excel file successfully saved to C:\Temp Folder Json Files\Video_Data_Export.xlsx
>>>
```

7. Done. The spreadsheet is saved in the same folder as the jsons.

Tips

- **Delete the dislikes column, it only works if you have json files you downloaded prior to 2022. On the homepage: Cells -> Delete -> Delete Sheet Columns.**
  I only coded it in because I have some left from before Youtube removed the dislike button. There might be some way to program ‘Return Youtube Dislike’ functionality into yt-dlp, I have no idea, but I doubt it.

- Each column has a small arrow at the bottom right of the top cell, this allows you to sort the rows Alphabetically, or numerically if applicable.


Views	Likes	Dislikes	Upload
331310			
488178			
625759			
653682			
811407			
876626			

- Double-click between the columns/rows to expand them to fit the longest string of text. Or drag them to fit how you'd like.

- If you want to change the formatting of the spreadsheet, go ahead. This can be done inside the script itself. You can just drag the script into an AI like Claude 3.5, and ask it to change formatting of the columns or header rows. Results may vary but AI excels at simple coding like this.

THE VIDEO AUTOCLIPPER

Ever wanted to automatically make up to thousands of clips out of any words someone said in a video (or thousands of videos)? Probably not, but it's very overpowered for making compilation videos.

I figured out a way to do it pretty easily; all you have to do is edit the clips together in a video editor at the end.

- **Remember, you'll need the video files you downloaded with YT-DLP to do this, and plenty of free space. You'll probably want a few hundred GBs of free space if you're doing a large Youtube channel (mainly to download the channel; making clips will be relatively low on space).**

- **Also you must install Python, OpenAI Whisper, and GitBash (on Windows) to download a few packages the scripts will use.**

- **You'll want to know how to use Python IDLE or another shell to run .py scripts, just follow the guide under the "SPECIAL TOOLS" section.**

This is a workflow I came up with by accident messing around with ChatGPT. There are 4 relatively simple scripts you'll run in this process, and it can be done in only about 6 steps on your end. **Once you get the hang of it, you can make mega-compilations of anything someone said in their videos.**

You must have around 8GB of graphics card VRAM to use OpenAI Whisper, and in this case it's the "Small" model. I haven't figured out how to get the Medium model working in Python yet; if you set it to medium the script will freeze while running.

There is one optional script you'll probably want to download to use this with YT-DLP:

- (The Mass 'Re-encode to h.264/AAC' Script)

  This is **crucial** if you're manually editing videos out of the clips, *UNLESS* you edit with DaVinci Resolve. *Adobe Premiere and Sony Vegas still don't support VP9 video formats, or OPUS audio, which is the YT-DLP default.*

There are 4 scripts you'll need to download to use this:

- (The OpenAI Whisper Transcription Script)

  You MUST install OpenAI Whisper first, **use this video.**

- (The Transcription 'Cleaner' Script)

- (The Word Detector Script)

- (The FFmpeg 'AutoClipper' Script)

I highly recommend that if you're making compilations of clips using this autoclipper, you download your yt-dlp videos with the upload date in front of the filename.

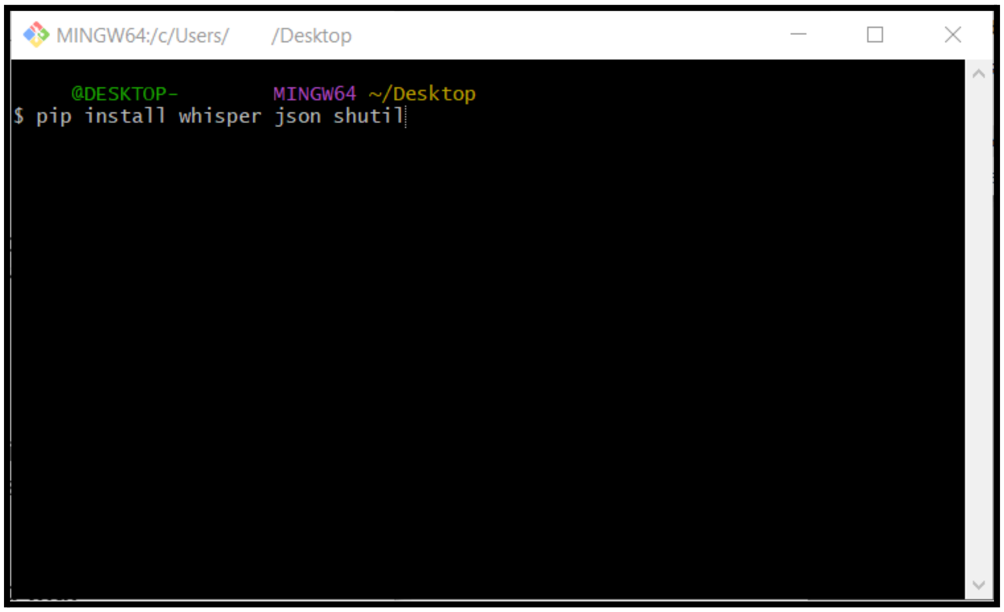
This is particularly useful if you want the compilations to be in chronological order.
It's very easy to set it up this way, because the transcripts are already timestamped in order, so even clips from the same video/day will be in order.

-o "[%(upload_date>%Y-%m-%d)s] %(title)s [%(id)s]"

1. Install the necessary packages the scripts will need, this can be easily done through GitBash (*follow guide under the "SPECIAL TOOLS" section*).

   pip install whisper json shutil

   Hit enter and wait.



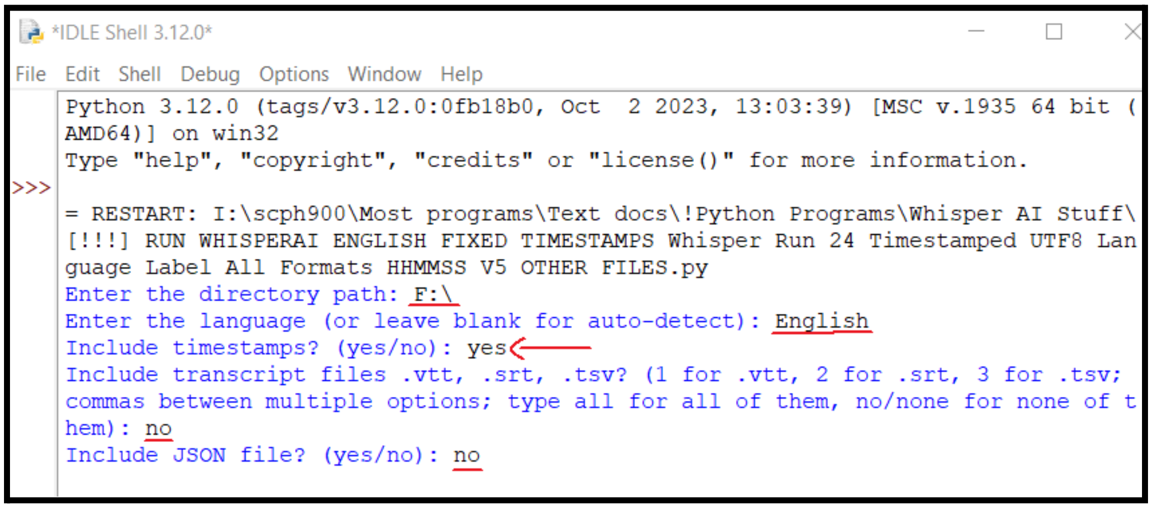
```
    @DESKTOP-        MINGW64 ~/Desktop
$ pip install whisper json shutil
```

2. **Transcribe video files you downloaded with yt-dlp.** This can be done with this Python script: follow these instructions to install OpenAI whisper on your computer, it's surprisingly simple.
   I also had different functions coded where you can choose whether to have subtitle files made alongside the .txt file transcription.

You *might* want to re-encode all the VP9/OPUS audio files you downloaded with yt-dlp, *provided you edit with Adobe Premiere/Sony Vegas*, before transcribing the files (scroll down to step 5). Or you can do it every time you have the clips generated, or just edit with another program like DaVinci Resolve.

- Make sure to specify the language (unless your videos are in varying languages), or it'll have to guess what language the video is and it can misdetect about 10% of the time, which is why it will output the language in the front of the filename if you don't specify the language.

- **Make sure to type yes on this option "Include timestamps? (yes/no):", otherwise this won't work. You are only going to use .txt transcription files, so unless you want subtitles, just type no two times after that.**



```
Python 3.12.0 (tags/v3.12.0:0fb18b0, Oct  2 2023, 13:03:39) [MSC v.1935 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: I:\scph900\Most programs\Text docs\!Python Programs\Whisper AI Stuff\
[!!!] RUN WHISPERAI ENGLISH FIXED TIMESTAMPS Whisper Run 24 Timestamped UTF8 Lan
guage Label All Formats HHMMSS V5 OTHER FILES.py
Enter the directory path: F:\
Enter the language (or leave blank for auto-detect): English
Include timestamps? (yes/no): yes
Include transcript files .vtt, .srt, .tsv? (1 for .vtt, 2 for .srt, 3 for .tsv;
commas between multiple options; type all for all of them, no/none for none of t
hem): no
Include JSON file? (yes/no): no
```

You will be left with .txt files, with timestamps, with the identical filename as the .mp4 files (don't rename the mp4 files, or txt files after running this script, or the Autoclipper won't function).

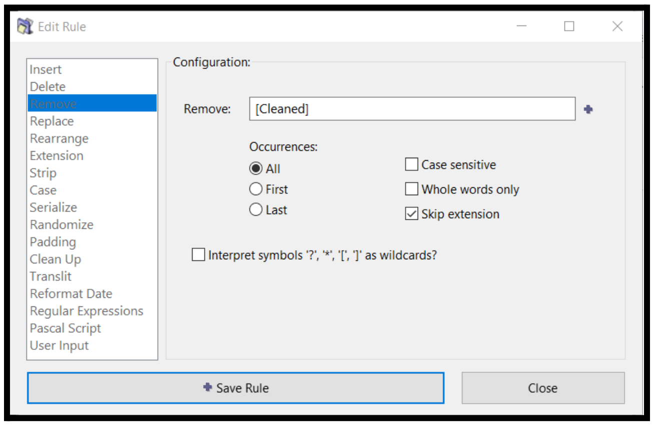
3. **Run the 'Cleaner' Script on the directory with the transcript files you generated.**

   This script will remove duplicate lines in the transcription generated by accident (most common with videos that have gaps in audio, and moments of silence; especially livestreams).

This is the main vulnerability of this method. OpenAI Whisper can sometimes completely ruin transcripts because of silence; there might be a solution but I haven't looked into it yet. **I would wager this 'Auto-Clipping' method is ~95% Accurate, so *almost* perfect.**

Check the filesize of your transcriptions after running this script, sometimes one or two will be only a few KB, effectively useless. I'd mark them elsewhere, or just delete because re-running it *might not* work.

You might want to remove the "[Cleaned]" part from the new files (after deleting the old ones, because they're inferior); you can do this easily and by the hundreds by downloading ReNamer and setting a "Remove" rule for " [Cleaned]".



Remove: [Cleaned]

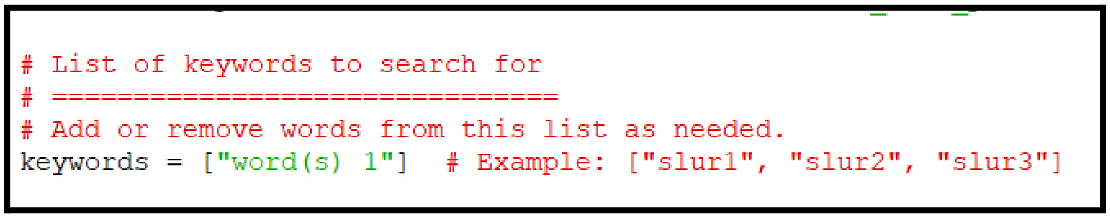
Occurrences: [✓ All] First Last

☐ Case sensitive
☐ Whole words only
☑ Skip extension

☐ Interpret symbols '?', '*', '[', ']' as wildcards?

4. **Run the Detections Script on the folder where you have the .txt transcription files.**
   The detection script searches for exact matches; you can search single word or exact quotes.

   Scroll down to the end of the Word Detector script to change it.



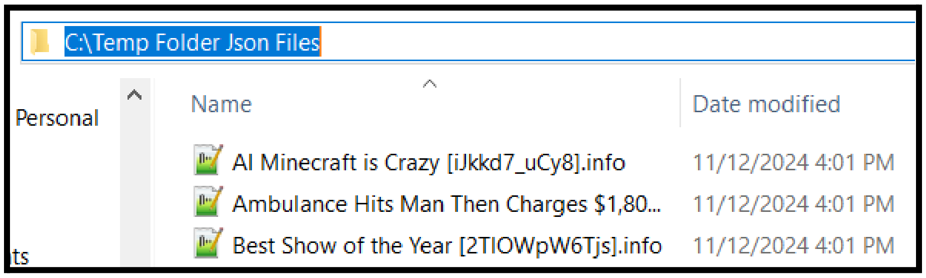
```
# List of keywords to search for
# ================================
# Add or remove words from this list as needed.
keywords = ["word(s) 1"]  # Example: ["slur1", "slur2", "slur3"]
```

You will include a comma between each additional word or series of words you're detecting: if you only want one, then there will be no comma.

Then run the script using F5, copy/paste the folder directory with your Transcription files, and you'll get .txt files with [Detections] in the filename. They'll only have the sentence where the word was detected.

5. **Copy the directories you have your .mp4 files into the "Clip Stream Vods BUFFER, Transcript Timestamps, in TONS of Directories 20" script. The transcript files will connect back to your .mp4 files using these directories.**

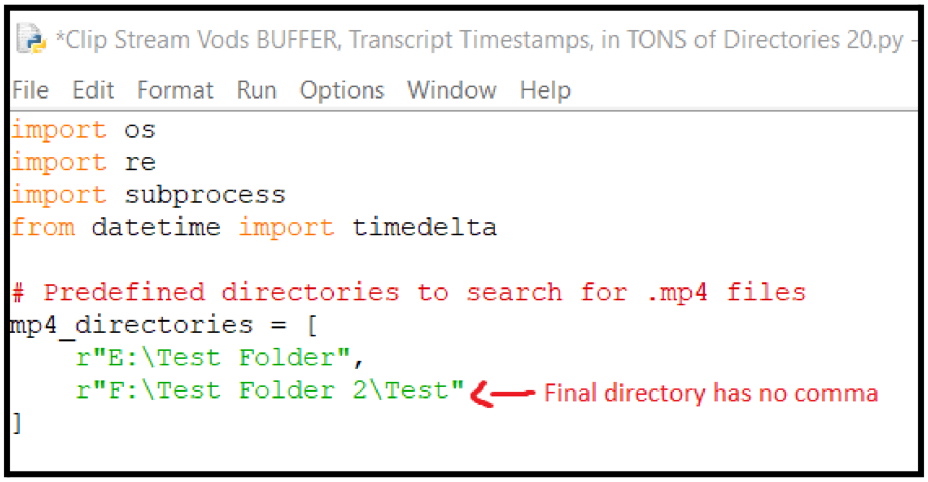
   All you have to do is click on the file path on the directory and paste the output between the quotation marks.



C:\Temp Folder Json Files

Name	Date modified
AI Minecraft is Crazy [iJkkd7_uCy8].info	11/12/2024 4:01 PM
Ambulance Hits Man Then Charges $1,80...	11/12/2024 4:01 PM
Best Show of the Year [2TIOWpW6Tjs].info	11/12/2024 4:01 PM

Be very careful to make sure there is no extra comma after the final directory you include; a comma must be included if you use more than one directory (you can use up to hundreds). **If you only use one directory, remove the first r"insert directory file path", which includes the comma at the end.**



*Clip Stream Vods BUFFER, Transcript Timestamps, in TONS of Directories 20.py
File Edit Format Run Options Window Help
Final directory has no comma

```
import os
import re
import subprocess
from datetime import timedelta

# Predefined directories to search for .mp4 files
mp4_directories = [
    r"E:\Test Folder",
    r"F:\Test Folder 2\Test"
]
```

Run the [FFmpeg Autoclipper Script](#) on the directory where you have the Transcript Detection files.

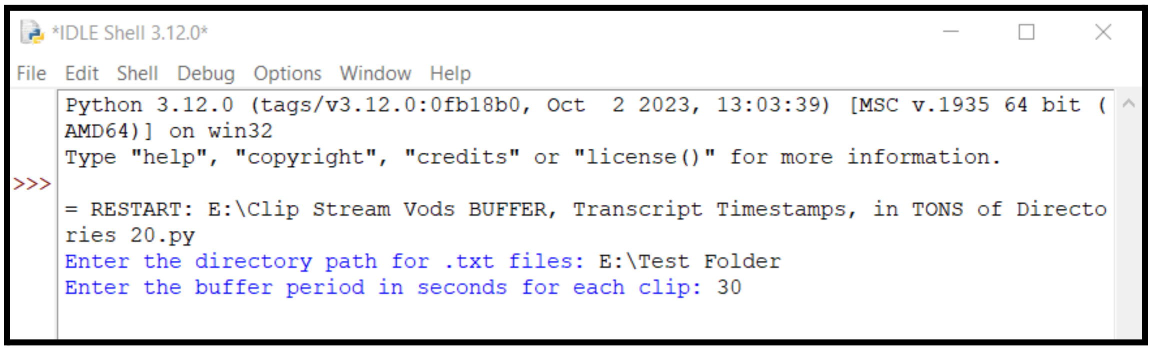
By default the script searches for .mp4 files in the directories you provide, but you can add more filetypes to be searched for.

You will be given the option to set the 'buffer period' for each clip; this refers to the amount of extra time surrounding the timestamp that the detection file used.

- **If your detection file has 00:01:15:000 - 00:02:15:000, and you set the buffer period for 60 seconds, the clip will start 60 seconds earlier <u>on both sides</u> at 00:00:15:000 - 00:03:15:000.**
  This is very useful for different tasks, and making sure clips have enough context.

- **You MUST have some level of a buffer period, otherwise the autoclipper will ruin many clips and not include enough context. I'd always recommend at least 5 seconds.**

  This makes it effectively impossible to not have to do at least some editing at the end, but that's an issue with video keyframes, and cannot be solved.
  If you're clipping something where you want a full conversation, a higher buffer period is ideal.



```
Python 3.12.0 (tags/v3.12.0:0fb18b0, Oct  2 2023, 13:03:39) [MSC v.1935 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: E:\Clip Stream Vods BUFFER, Transcript Timestamps, in TONS of Directo
ries 20.py
Enter the directory path for .txt files: E:\Test Folder
Enter the buffer period in seconds for each clip: 30
```

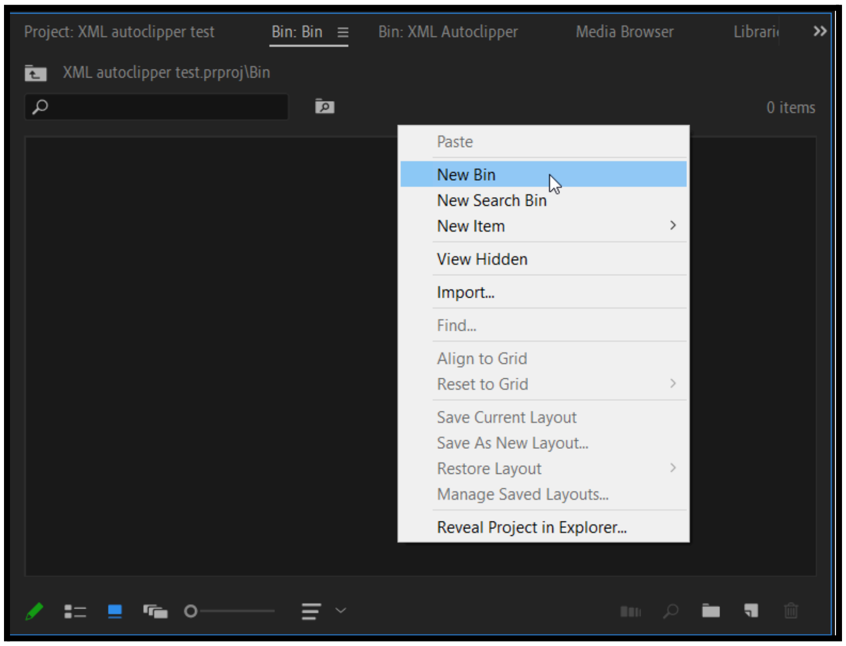
Just wait depending on how many clips you're making, and you'll get .mp4 files created for every video you clipped. Might take hours depending on the bitrate of the video files, buffer period, power of your CPU/GPU, and number of individual detections per .txt file.
By the way, each line in the detection file gets clipped, so make sure to not use the original OpenAI transcript files, or it will make hundreds/thousands of clips.

6. **If you haven't already converted the video files you downloaded with yt-dlp to h.264/AAC, then you'll need to do it before you start editing (unless you use DaVinci Resolve to edit). If you use Adobe Premiere or Sony Vegas this is mandatory, or the video files are un-editable.**

Mass-convert the VP9/OPUS files into h.264/AAC using the "RE-ENCODE ALL OPUS MP4 IN A DIRECTORY" script.

7. **(Optional, recommended workflow for Adobe Premiere)**
[If you don't have it, just find a reputable crack so you don't have to pay; https://www.reddit.com/r/GenP]. M0nkrus Adobe cracks are well known on Reddit and very simple to install.

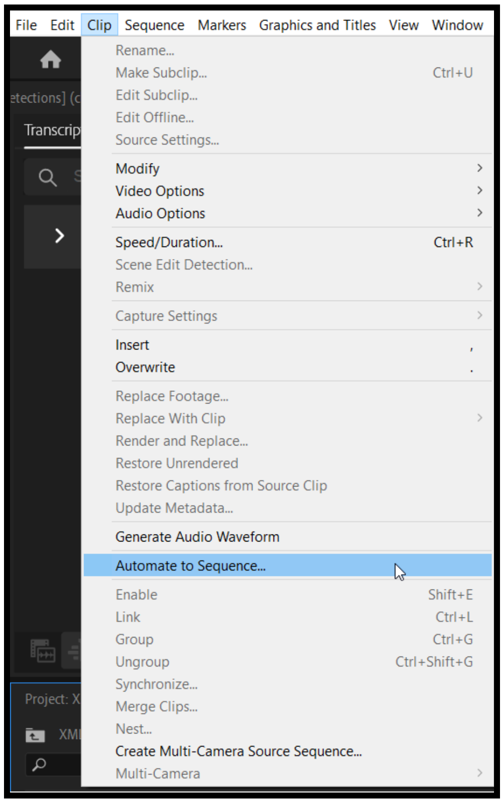
    If you're editing these clips into a single video in Adobe, this can be automated by importing all of the files at once into a Bin in the project panel.



- **Back in the directory with your clip files, sort the folder by filetype to separate the .mp4s from the .txt files in the directory, move them to a new one, and re-sort them by Name.**

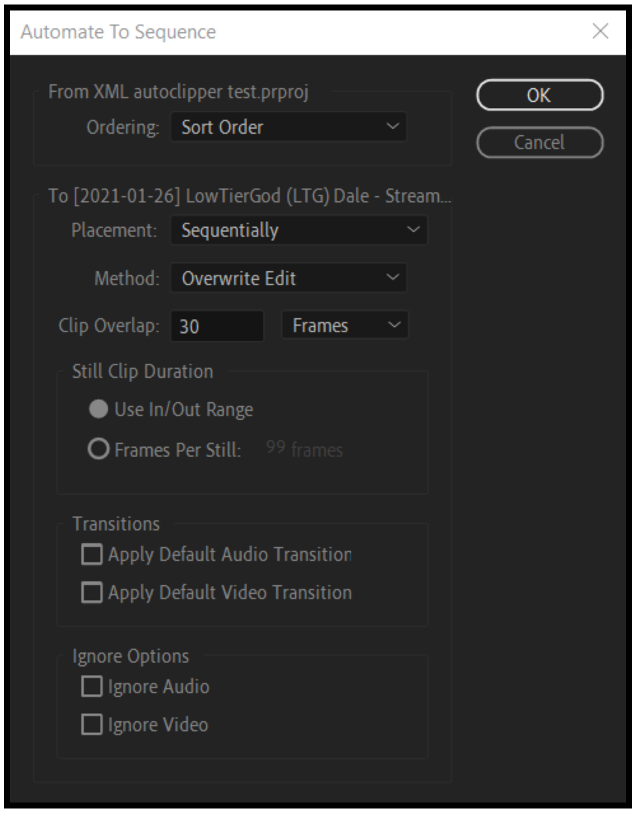
- **Drag all of the clip files into the bin; not the timeline.**

- **CTRL-A to select them all, <u>make sure there is an empty timeline</u> already existing (otherwise the next selection is greyed out), and then select "Automate to Sequence."**



- **Set the ordering as "Sort Order" if you have the already organized chronologically (if they're auto-dated).**

  Every clip will be in order instantly, so all you have to do next is edit the clips down and export.


Automate To Sequence

From XML autoclipper test.prproj

Ordering: Sort Order

To [2021-01-26] LowTierGod (LTG) Dale - Stream...

Placement: Sequentially

Method: Overwrite Edit

Clip Overlap: 30 Frames

Still Clip Duration

- [●] Use In/Out Range
- [ ] Frames Per Still: 99 frames

Transitions

- [ ] Apply Default Audio Transition
- [ ] Apply Default Video Transition

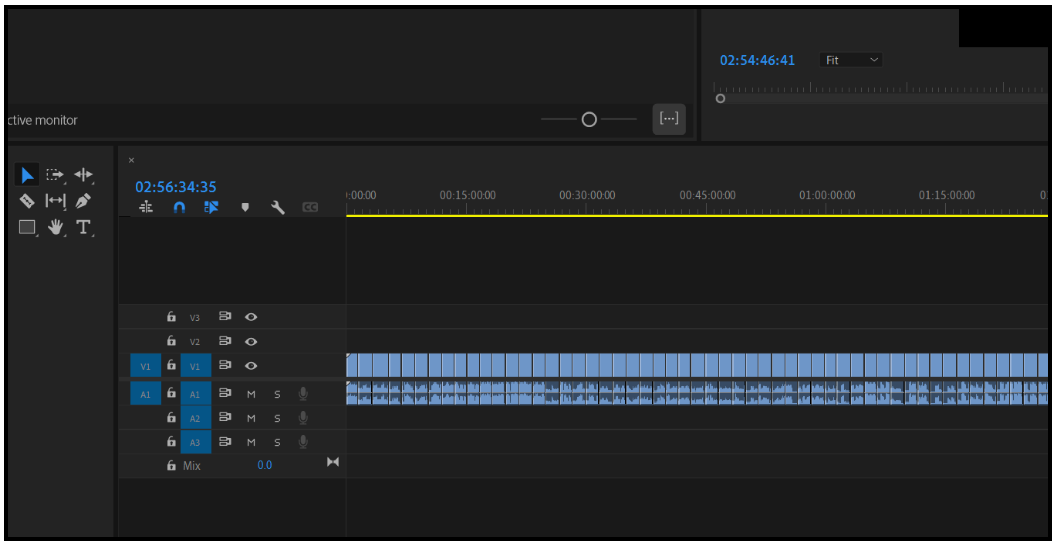
Ignore Options

- [ ] Ignore Audio
- [ ] Ignore Video

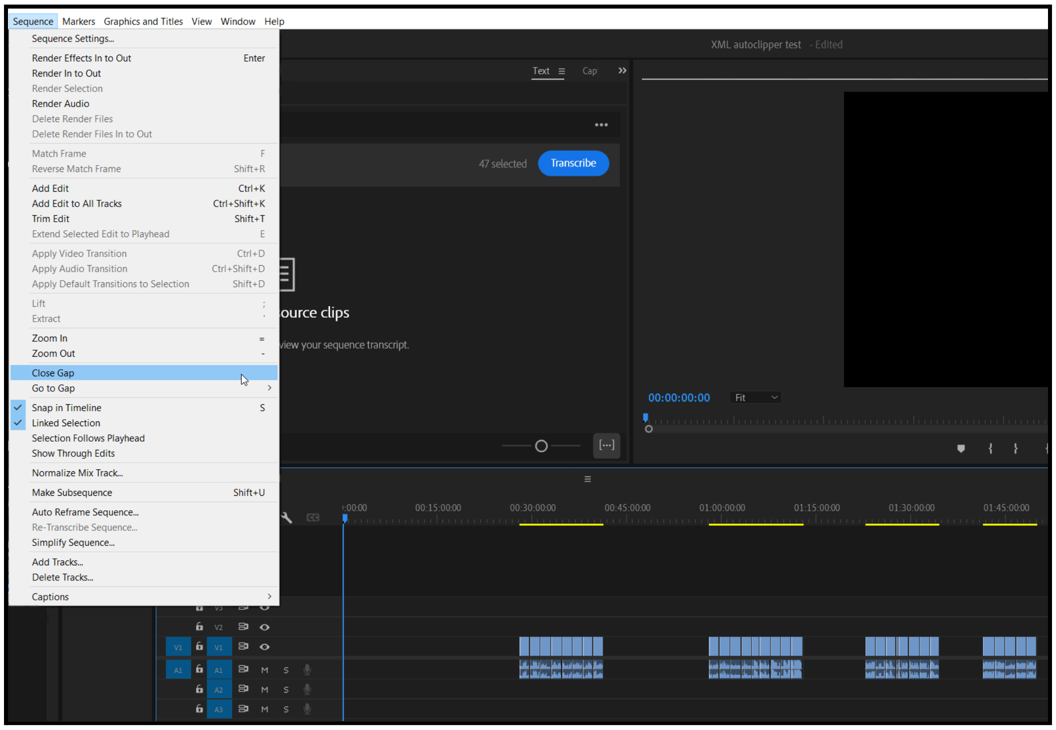
- If you're editing clips to be watchable (i.e. you used any sort of buffer period in the autoclipper script), just zoom in and use the "C" key to trim each clip individually and delete the filler surrounding the clip.

  Tip: Press the "L" Key and then Multiple "Shift+L" presses to increase preview playback speed (two L presses is a little too quick, so one L and then 4-5 Shift-L's is ideal). There is no other way to speed up the speed like this in Premiere, without editing the actual clip.

  Incredibly useful when having to sit through each clip to find new start and end times.



- Finally, while you're editing the clips down, to re-join them into a unified timeline just select them all, click Sequence on the Menu, and "Close Gap."



Notes

Theoretically, having 0 buffer period would let you have the exact timestamp for the sentence the word is contained in, but this doesn't work in practice due to FFmpeg having issues with trimming off timestamps. **You'll want at least 4 seconds for a buffer period, or a clip could end up with a shorter clip than the actual timestamp it should've clipped.**

There might be some method to 'intelligently' trim the clips automatically using some external script, who knows, but you'll always have to do a little editing for each clip depending on your buffer period, unless you're okay with some gaps.

I've explored this concept a little with AI, but it seems it wouldn't work due to OpenAI Whisper having entire sentences with timestamp ranges. Specifically, it uses "sentence-level timestamps" instead of "word-level timestamps."

You actually *can* set word-level timestamps as a function in OpenAI Whisper. So with a modified script you could have detections for an exact word with exact timestamps.

But again, you have to have a buffer period, which makes it not much of an improvement over the existing script. Effectively, all that would change is it's clipping around a word instead of a whole sentence, meaning you'd need a larger buffer period. No advantage.

If this concept even worked, it would also mess up compatibility with more than a single word because words won't be right next to each other anymore, instead being on separate lines.

The reason a buffer period is required is because of 'keyframes' in videos. All video files have keyframes and FFmpeg cannot avoid this.

You cannot clip less than 5 seconds using FFmpeg (vast majority of the time), so if you're trying to clip **exact** words, it can't be done using this method.

SPLIT VIDEO EVENLY INTO TWO (Superior to Re-Rendering)

Remove IPs From yt-dlp info.jsons

This script will run on a directory of your choosing, and remove all instances of IP addresses stored in .json files made by YT-DLP.

Download

Delete IPs in Directory of Jsons (newest5.24.25).py

This was mentioned earlier, but I don’t advise uploading the .info.json files anywhere without either using a VPN, or running this script to erase them.

Transcribe All Video Files Automatically (OpenAI Whisper)

Refer to the “VIDEO AUTOCLIPPER” section, the setup for it is the same as this.

You can follow the instructions in this video first:
How to Install & Use Whisper AI Voice to Text

But in this case, you want to install the prerequisites and *avoid* using WhisperAI as shown in the video; my python version operates much more effectively than manually through the command line.
Unlike through the command line, my script can run iteratively on any number of files without needing to reference them manually, and lets you choose whether to save timestamps, subtitles, and jsons.

Convert All VP9/OPUS Files into H264 (for video editing)

This is tailored more towards people that use the typical video editors like Adobe or Vegas, these programs somehow even in 2025 don’t support the VP9 video codec and OPUS audio codec. **More context in the “Disclaimer for Video Editors” section.**

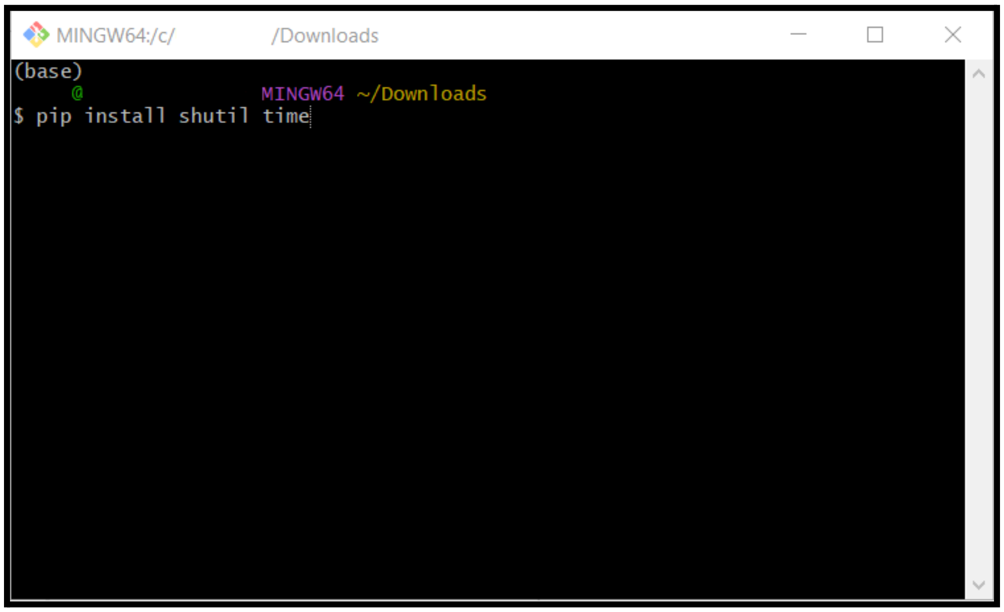
You can use DaVinci Resolve if you don’t want to bother doing this, but it's relatively simple.

This is also useful to make mega compilation videos of a youtuber, clips, etc. Otherwise you, more or less, have to use DaVinci Resolve or another editor.

(note: improve this)

Required Packages:

- **shutil**
- **time**



```
(base)
      @                 MINGW64 ~/Downloads
$ pip install shutil time
```

Download

https://drive.google.com/file/d/1t9XWsfznf_GK1LO_IitGRA_ivW7qvAkC

Say you downloaded an entire youtube channel's worth of videos, but you can't drag them into a video editor because Adobe is shit.

All you must do is run this script on the directory of video files you want to turn into usable content, and it will automatically separate VP9 videos into a new subfolder, and within that sub-folder it will re-encode those to h264 (into another subfolder).

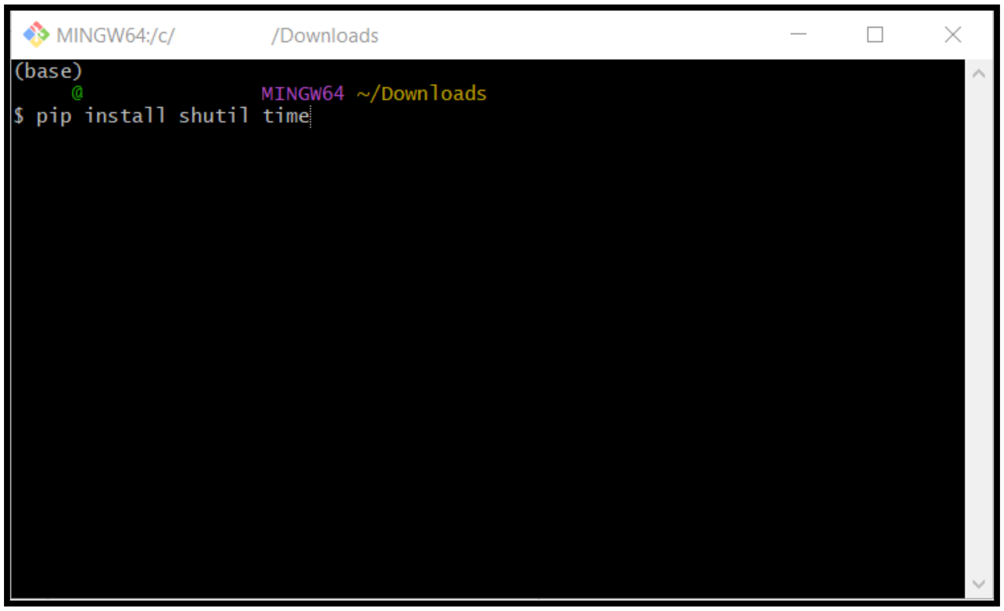
Convert All WEBP/JFIF To JPG Files

Simple script that lets you **replace** all .webp and .JFIF files in a directory with a converted .JPG. If you don't want a particular file converted or want .png as the output file, those can be set in the script yourself.

Currently, it will create files with a new Date Modified value, so if you sort your images by date modified, this will put them at the top.

Required Packages:

- **shutil**



```
(base)
      @                MINGW64 ~/Downloads
$ pip install shutil time
```

<u>Download</u>

https://drive.google.com/file/d/1FVo6jE6iZtLtZv1eWI3ESjgWNyQmYfhy

Run the script, paste the directory you want to convert, and it will work immediately.

Add Dates to mp4s Using info.json Files (etc.)

Perfect for chronological sorting

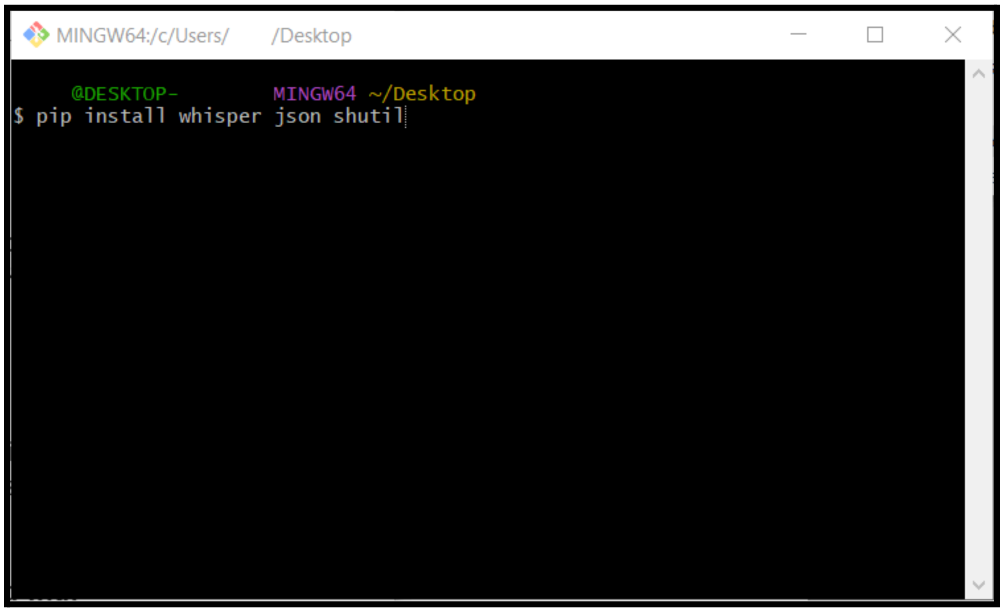
If you download the info.json files when using yt-dlp, youtube and rumble saves the video upload_date inside the file. This can easily be used to rename your video files to sort chronologically, like this:

[2025-04-01] jgiojosdg [etjsp0djgs].mp4

This is a nice option to have, as it's more customizable to your own interests, compared to setting a fixed output command like *-o "[%(upload_date>%Y-%m-%d)s] %(title)s [%(id)s]"*.

<u>Required Packages:</u>

- **json**

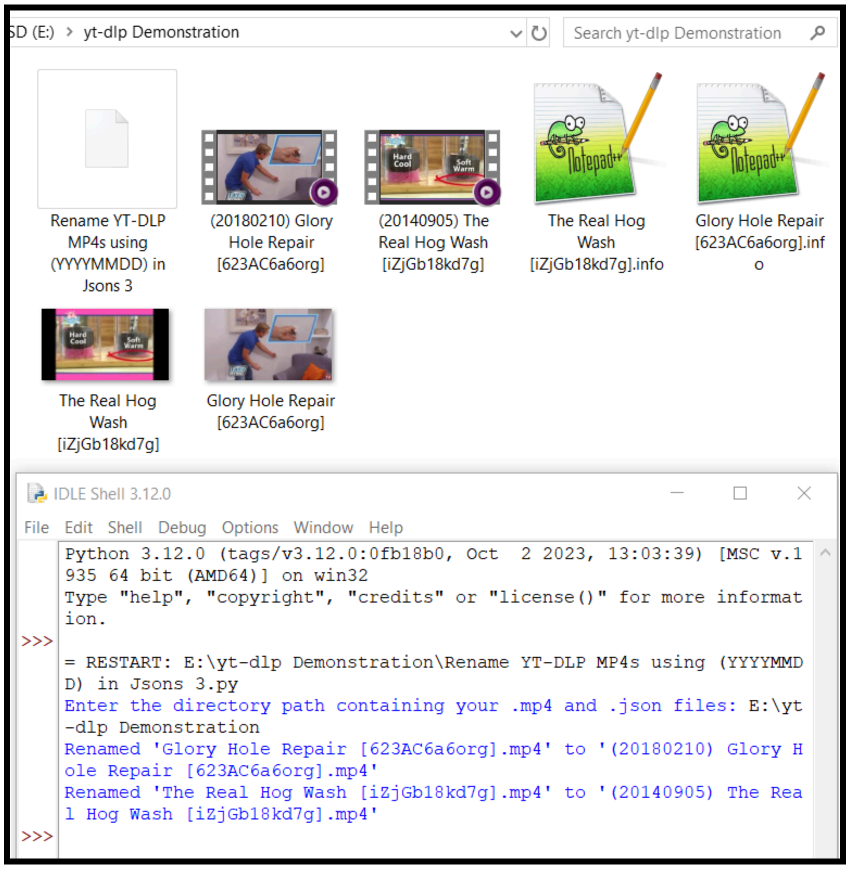


```
     @DESKTOP-        MINGW64 ~/Desktop
$ pip install whisper json shutil
```

I've made several variants of this script for my use cases, so just pick the one you want. You can also add more file extensions, or replace them, within the script to apply to certain file types (for example, mp3/wav files for audio).

1. Rename YT-DLP **MP4s** using (YYYYMMDD):

   <u>Download</u>: [Rename YT-DLP MP4s using (YYYYMMDD) in Jsons 3.py]()

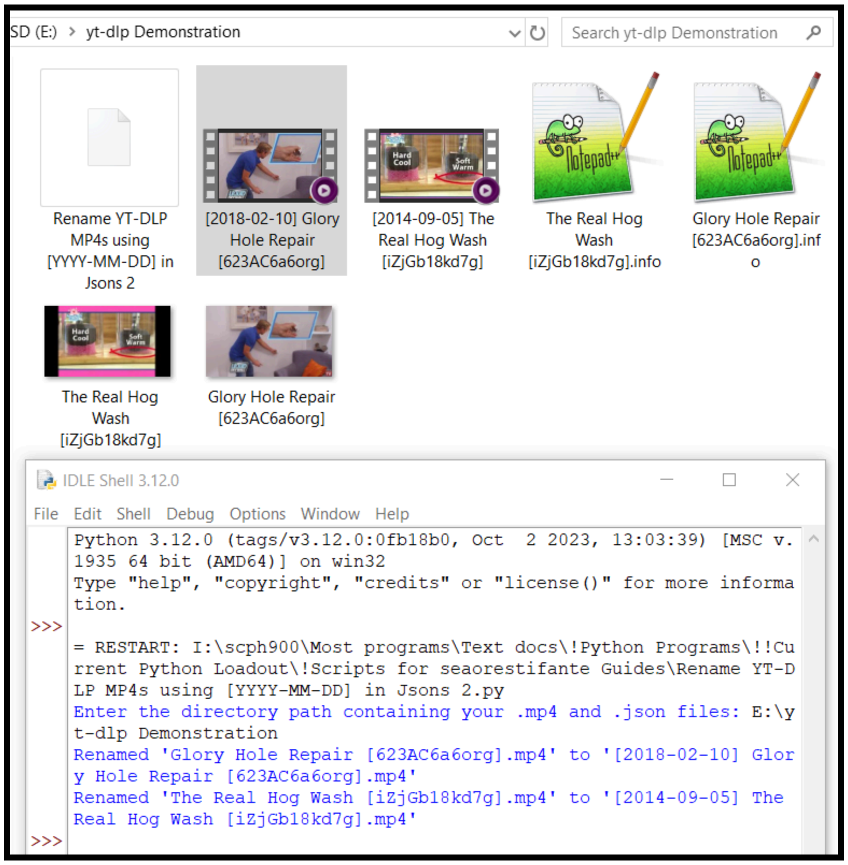

SD (E:) > yt-dlp Demonstration
Search yt-dlp Demonstration
Rename YT-DLP MP4s using (YYYYMMDD) in Jsons 3
(20180210) Glory Hole Repair [623AC6a6org]
(20140905) The Real Hog Wash [iZjGb18kd7g]
The Real Hog Wash [iZjGb18kd7g].info
Glory Hole Repair [623AC6a6org].info
The Real Hog Wash [iZjGb18kd7g]
Glory Hole Repair [623AC6a6org]
IDLE Shell 3.12.0
File Edit Shell Debug Options Window Help

```
Python 3.12.0 (tags/v3.12.0:0fb18b0, Oct  2 2023, 13:03:39) [MSC v.1
935 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more informat
ion.
>>>
= RESTART: E:\yt-dlp Demonstration\Rename YT-DLP MP4s using (YYYYMMD
D) in Jsons 3.py
Enter the directory path containing your .mp4 and .json files: E:\yt
-dlp Demonstration
Renamed 'Glory Hole Repair [623AC6a6org].mp4' to '(20180210) Glory H
ole Repair [623AC6a6org].mp4'
Renamed 'The Real Hog Wash [iZjGb18kd7g].mp4' to '(20140905) The Rea
l Hog Wash [iZjGb18kd7g].mp4'
>>>
```

- **Format (20250401) FILENAME.mp4**

2. Rename YT-DLP **MP4s** using [YYYY-MM-DD]:

Download: Rename YT-DLP MP4s using [YYYY-MM-DD] in Jsons 2.py


SD (E:) > yt-dlp Demonstration
Search yt-dlp Demonstration
Rename YT-DLP MP4s using [YYYY-MM-DD] in Jsons 2
[2018-02-10] Glory Hole Repair [623AC6a6org]
[2014-09-05] The Real Hog Wash [iZjGb18kd7g]
The Real Hog Wash [iZjGb18kd7g].info
Glory Hole Repair [623AC6a6org].info
The Real Hog Wash [iZjGb18kd7g]
Glory Hole Repair [623AC6a6org]
IDLE Shell 3.12.0
File Edit Shell Debug Options Window Help

```
Python 3.12.0 (tags/v3.12.0:0fb18b0, Oct  2 2023, 13:03:39) [MSC v.
1935 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more informa
tion.
>>>
= RESTART: I:\scph900\Most programs\Text docs\!Python Programs\!!Cu
rrent Python Loadout\!Scripts for seaorestifante Guides\Rename YT-D
LP MP4s using [YYYY-MM-DD] in Jsons 2.py
Enter the directory path containing your .mp4 and .json files: E:\y
t-dlp Demonstration
Renamed 'Glory Hole Repair [623AC6a6org].mp4' to '[2018-02-10] Glor
y Hole Repair [623AC6a6org].mp4'
Renamed 'The Real Hog Wash [iZjGb18kd7g].mp4' to '[2014-09-05] The
Real Hog Wash [iZjGb18kd7g].mp4'
>>>
```

- **Format [2025-04-01] FILENAME.mp4**

```
# Process each mp4 file
for file in os.listdir(directory):
    if file.endswith('.mp3'):
```

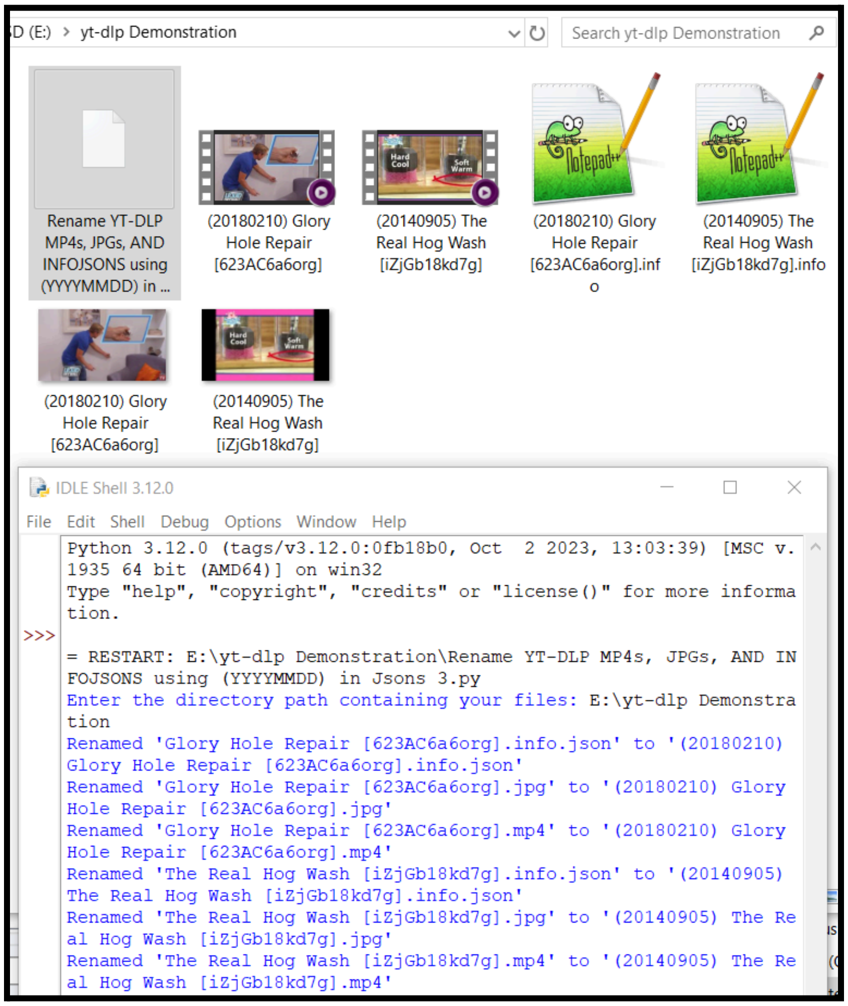
You can change which files it applies to here.

The following scripts apply to all the file types I typically encounter. It also renames the json file itself.

---

3. Rename YT-DLP **MP4s, JPGs, AND INFOJSONS** using (YYYY-MM-DD):

Download: Rename YT-DLP MP4s, JPGs, AND INFOJSONS using (YYYYMMDD) in Jsons 3.py

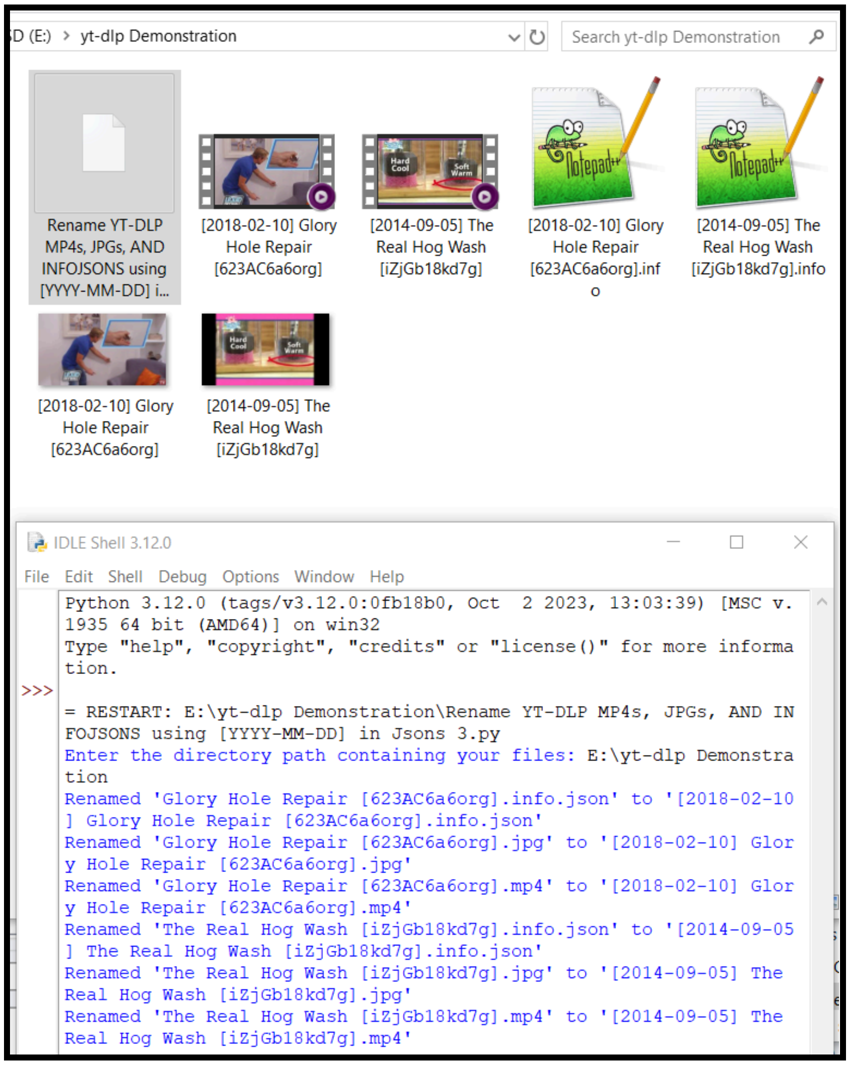

D (E:) > yt-dlp Demonstration
Search yt-dlp Demonstration
Rename YT-DLP MP4s, JPGs, AND INFOJSONS using (YYYYMMDD) in ...
(20180210) Glory Hole Repair [623AC6a6org]
(20140905) The Real Hog Wash [iZjGb18kd7g]
(20180210) Glory Hole Repair [623AC6a6org].info
(20140905) The Real Hog Wash [iZjGb18kd7g].info
(20180210) Glory Hole Repair [623AC6a6org]
(20140905) The Real Hog Wash [iZjGb18kd7g]
IDLE Shell 3.12.0
File Edit Shell Debug Options Window Help

```
Python 3.12.0 (tags/v3.12.0:0fb18b0, Oct  2 2023, 13:03:39) [MSC v.
1935 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more informa
tion.
>>>
= RESTART: E:\yt-dlp Demonstration\Rename YT-DLP MP4s, JPGs, AND IN
FOJSONS using (YYYYMMDD) in Jsons 3.py
Enter the directory path containing your files: E:\yt-dlp Demonstra
tion
Renamed 'Glory Hole Repair [623AC6a6org].info.json' to '(20180210)
Glory Hole Repair [623AC6a6org].info.json'
Renamed 'Glory Hole Repair [623AC6a6org].jpg' to '(20180210) Glory
Hole Repair [623AC6a6org].jpg'
Renamed 'Glory Hole Repair [623AC6a6org].mp4' to '(20180210) Glory
Hole Repair [623AC6a6org].mp4'
Renamed 'The Real Hog Wash [iZjGb18kd7g].info.json' to '(20140905)
The Real Hog Wash [iZjGb18kd7g].info.json'
Renamed 'The Real Hog Wash [iZjGb18kd7g].jpg' to '(20140905) The Re
al Hog Wash [iZjGb18kd7g].jpg'
Renamed 'The Real Hog Wash [iZjGb18kd7g].mp4' to '(20140905) The Re
al Hog Wash [iZjGb18kd7g].mp4'
```

- **Format (20250401) FILENAME.mp4**

4. Rename YT-DLP **MP4s, JPGs, AND INFOJSONS** using [YYYY-MM-DD]:

   Download: Rename YT-DLP MP4s using [YYYY-MM-DD] in Jsons 2.py


D (E:) > yt-dlp Demonstration
Search yt-dlp Demonstration
Rename YT-DLP MP4s, JPGs, AND INFOJSONS using [YYYY-MM-DD] i...
[2018-02-10] Glory Hole Repair [623AC6a6org]
[2014-09-05] The Real Hog Wash [iZjGb18kd7g]
[2018-02-10] Glory Hole Repair [623AC6a6org].info
[2014-09-05] The Real Hog Wash [iZjGb18kd7g].info
[2018-02-10] Glory Hole Repair [623AC6a6org]
[2014-09-05] The Real Hog Wash [iZjGb18kd7g]
IDLE Shell 3.12.0
File Edit Shell Debug Options Window Help

```
Python 3.12.0 (tags/v3.12.0:0fb18b0, Oct  2 2023, 13:03:39) [MSC v.
1935 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more informa
tion.
>>>
= RESTART: E:\yt-dlp Demonstration\Rename YT-DLP MP4s, JPGs, AND IN
FOJSONS using [YYYY-MM-DD] in Jsons 3.py
Enter the directory path containing your files: E:\yt-dlp Demonstra
tion
Renamed 'Glory Hole Repair [623AC6a6org].info.json' to '[2018-02-10
] Glory Hole Repair [623AC6a6org].info.json'
Renamed 'Glory Hole Repair [623AC6a6org].jpg' to '[2018-02-10] Glor
y Hole Repair [623AC6a6org].jpg'
Renamed 'Glory Hole Repair [623AC6a6org].mp4' to '[2018-02-10] Glor
y Hole Repair [623AC6a6org].mp4'
Renamed 'The Real Hog Wash [iZjGb18kd7g].info.json' to '[2014-09-05
] The Real Hog Wash [iZjGb18kd7g].info.json'
Renamed 'The Real Hog Wash [iZjGb18kd7g].jpg' to '[2014-09-05] The
Real Hog Wash [iZjGb18kd7g].jpg'
Renamed 'The Real Hog Wash [iZjGb18kd7g].mp4' to '[2014-09-05] The
Real Hog Wash [iZjGb18kd7g].mp4'
```

- **Format [2025-04-01] FILENAME.mp4**

```
# Step 2: Rename all relevant files
for file in os.listdir(directory):
    if file.endswith(('.mp4', '.jpg', '.png', '.info.json', '.info.txt', '.info.html')):
```

You can change which files it applies to here.

- *.info.txt is a file created by my "EXTRACT COMMENTS FROM JSON" script in the next section.* ***It contains all the youtube comments in a raw text format.***

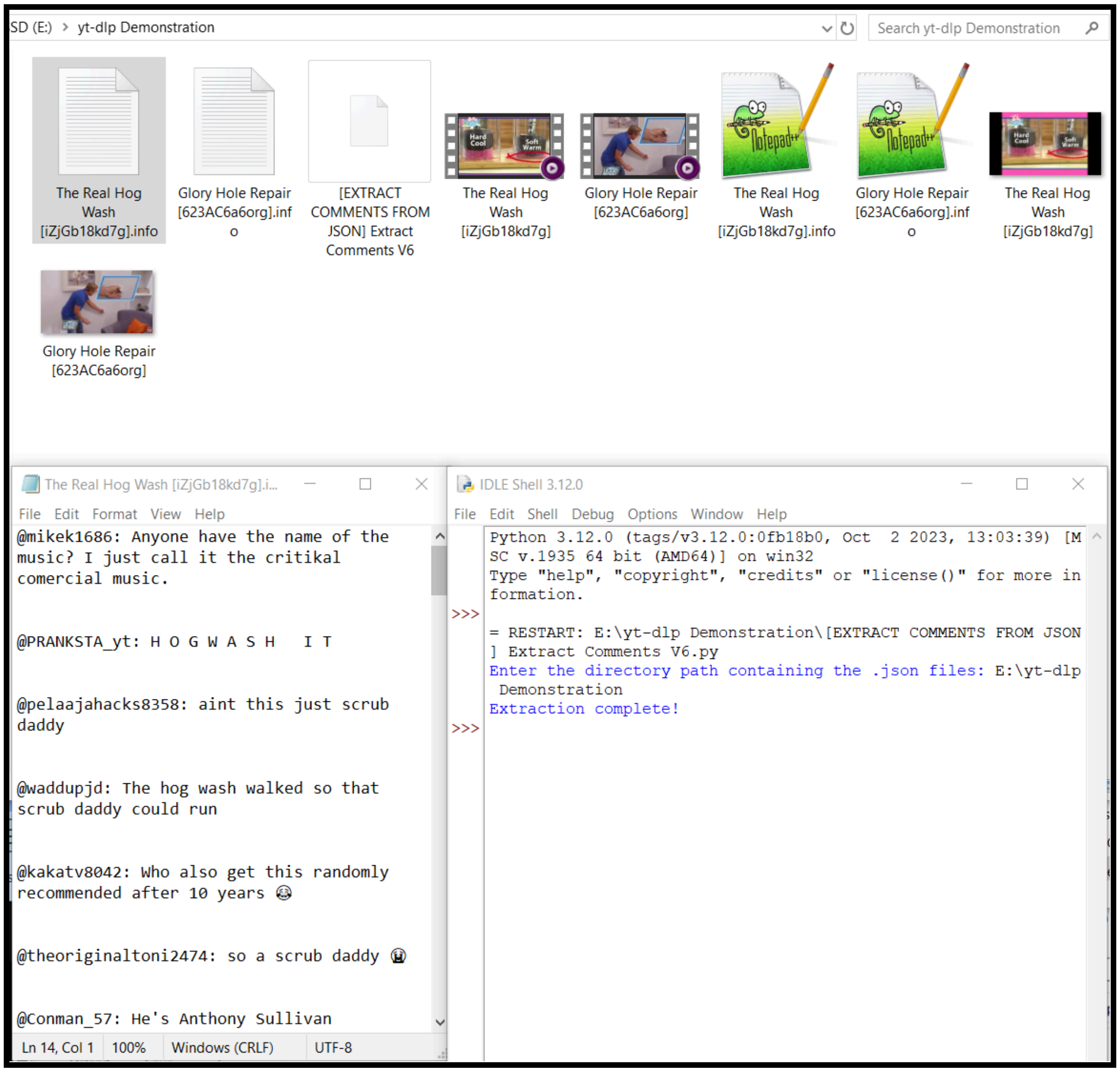
- *.info.html in a file created by my "Render YouTube Comments as an HTML Webpage" script. **It contains all the metadata + youtube comments in a formatted fake youtube page.***

Extract Youtube Comments from info.jsons as an .info.txt.

This was the predecessor to the "*Render YT-DLP Jsons as Fake Youtube Page*" script, which does it much better than this.

All this script does is extract the comments from the .info.json files yt-dlp makes, and makes a .txt file with each comment **(the channel name, and the comment, in no particular order)**.

Download: [EXTRACT COMMENTS FROM JSON] Extract Comments V6.py


SD (E:) > yt-dlp Demonstration
Search yt-dlp Demonstration
The Real Hog Wash [iZjGb18kd7g].info
Glory Hole Repair [623AC6a6org].info
[EXTRACT COMMENTS FROM JSON] Extract Comments V6
The Real Hog Wash [iZjGb18kd7g]
Glory Hole Repair [623AC6a6org]
The Real Hog Wash [iZjGb18kd7g].info
Glory Hole Repair [623AC6a6org].info
The Real Hog Wash [iZjGb18kd7g]
Glory Hole Repair [623AC6a6org]
The Real Hog Wash [iZjGb18kd7g].i...
File Edit Format View Help
Ln 14, Col 1 100% Windows (CRLF) UTF-8
IDLE Shell 3.12.0
File Edit Shell Debug Options Window Help

```
@mikek1686: Anyone have the name of the
music? I just call it the critikal
comercial music.

@PRANKSTA_yt: H O G W A S H   I T

@pelaajahacks8358: aint this just scrub
daddy

@waddupjd: The hog wash walked so that
scrub daddy could run

@kakatv8042: Who also get this randomly
recommended after 10 years 😂

@theoriginaltoni2474: so a scrub daddy 😱

@Conman_57: He's Anthony Sullivan
```

```
Python 3.12.0 (tags/v3.12.0:0fb18b0, Oct  2 2023, 13:03:39) [M
SC v.1935 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more in
formation.
>>>
= RESTART: E:\yt-dlp Demonstration\[EXTRACT COMMENTS FROM JSON
] Extract Comments V6.py
Enter the directory path containing the .json files: E:\yt-dlp
 Demonstration
Extraction complete!
>>>
```

Merge them into a Single Comment File (for an <u>entire</u> channel)

There are some interesting use cases for this, such as extracting the comments of a video, or an entire channel, and finding text patterns. Or searching for commenters.

I have another script which can combine all the .info.txt files into a single one. I can use that combined comment file, with an AI chatbot, to gauge the 'average'

comment on any youtube channel, or draw conclusions about what people talk about on a video/channel.

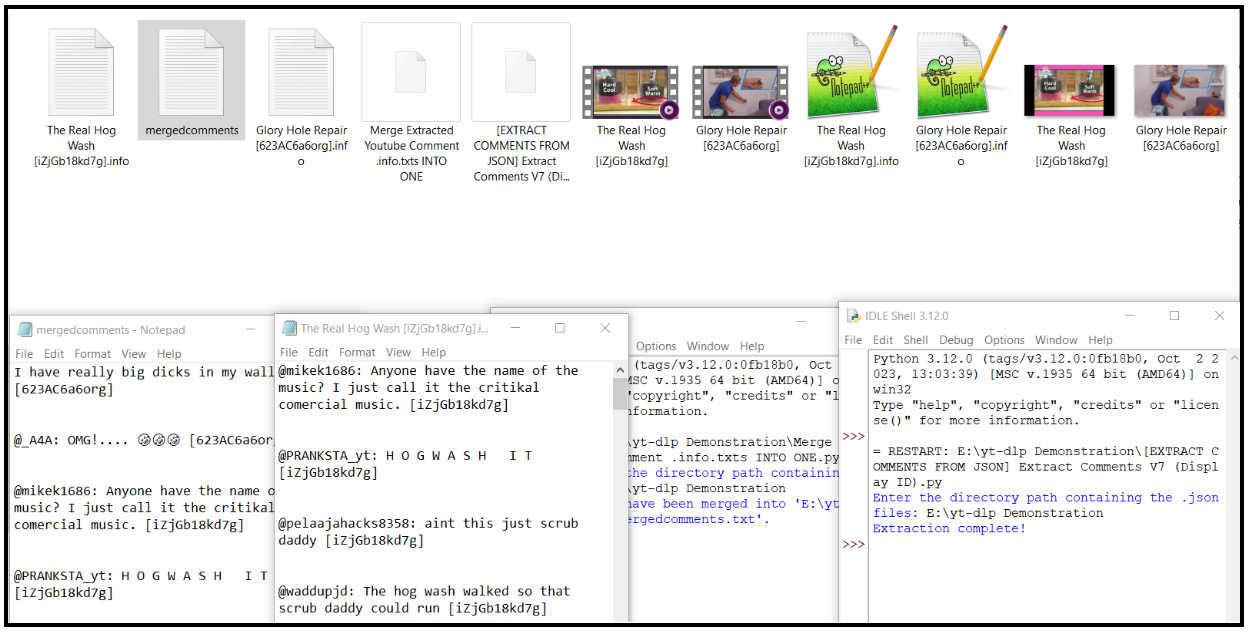
Download

[Merge Extracted Youtube Comment .info.txts INTO ONE.py]

On top of that, you may want this version of the Comment Extraction script, which appends the video ID at the end of each comment, so you can actually trace each comment (to the video it was posted on) after you merge the comments into a single file.

Download

[[EXTRACT COMMENTS FROM JSON] Extract Comments V7 (Display ID).py]


The Real Hog Wash [iZjGb18kd7g].info
mergedcomments
Glory Hole Repair [623AC6a6org].info
Merge Extracted Youtube Comment .info.txts INTO ONE
[EXTRACT COMMENTS FROM JSON] Extract Comments V7 (Di...
The Real Hog Wash [iZjGb18kd7g]
Glory Hole Repair [623AC6a6org]
The Real Hog Wash [iZjGb18kd7g].info
Glory Hole Repair [623AC6a6org].info
The Real Hog Wash [iZjGb18kd7g]
Glory Hole Repair [623AC6a6org]
mergedcomments - Notepad
File Edit Format View Help
The Real Hog Wash [iZjGb18kd7g].i...
File Edit Format View Help
Options Window Help
IDLE Shell 3.12.0
File Edit Shell Debug Options Window Help

```
I have really big dicks in my wall
[623AC6a6org]


@_A4A: OMG!.... 😂😂😂 [623AC6a6or


@mikek1686: Anyone have the name o
music? I just call it the critikal
comercial music. [iZjGb18kd7g]


@PRANKSTA_yt: H O G W A S H   I T
[iZjGb18kd7g]
```

```
@mikek1686: Anyone have the name of the
music? I just call it the critikal
comercial music. [iZjGb18kd7g]


@PRANKSTA_yt: H O G W A S H   I T
[iZjGb18kd7g]


@pelaajahacks8358: aint this just scrub
daddy [iZjGb18kd7g]


@waddupjd: The hog wash walked so that
scrub daddy could run [iZjGb18kd7g]
```

```
 (tags/v3.12.0:0fb18b0, Oct
MSC v.1935 64 bit (AMD64)] o
'copyright", "credits" or "l
nformation.

\yt-dlp Demonstration\Merge
mment .info.txts INTO ONE.py
the directory path containin
\yt-dlp Demonstration
have been merged into 'E:\yt
ergedcomments.txt'.
```

```
Python 3.12.0 (tags/v3.12.0:0fb18b0, Oct  2 2
023, 13:03:39) [MSC v.1935 64 bit (AMD64)] on
win32
Type "help", "copyright", "credits" or "licen
se()" for more information.
>>>
= RESTART: E:\yt-dlp Demonstration\[EXTRACT C
OMMENTS FROM JSON] Extract Comments V7 (Displ
ay ID).py
Enter the directory path containing the .json
files: E:\yt-dlp Demonstration
Extraction complete!
>>>
```

Search yt-dlp info.jsons for Comments from a Specific Author

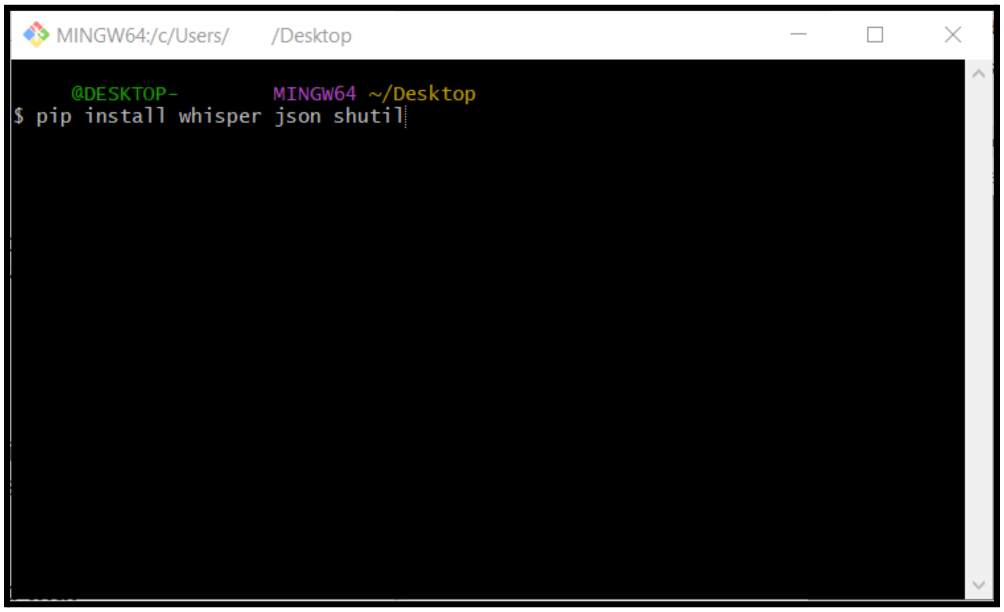
Here is a script which can find matches for any one commenter, in a single comment section (one .info.json with –write-comments enabled), or an entire youtube channel.

You must specify the @ handle before the username you're searching for, unless the .info.json was created before youtube replaced commenter names with handles in 2024.

(note: improve this)

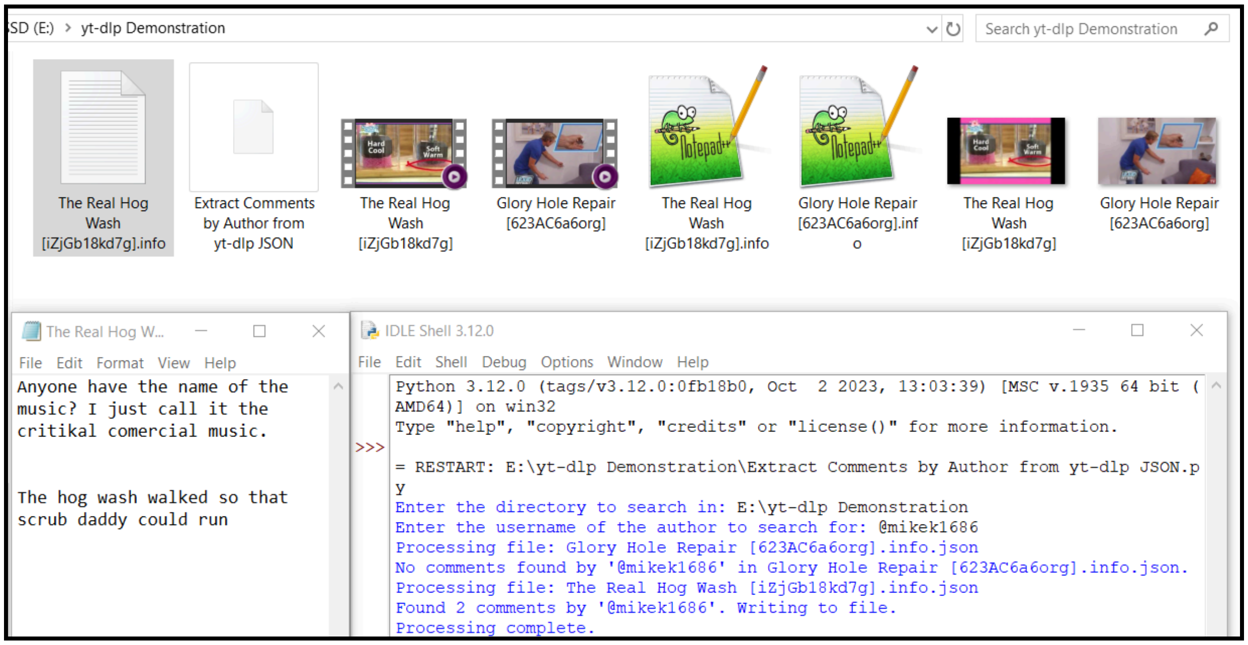
Required Packages:

- **json**



```
MINGW64:/c/Users/      /Desktop
     @DESKTOP-        MINGW64 ~/Desktop
$ pip install whisper json shutil
```

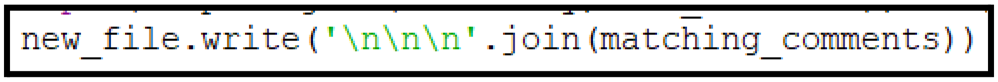
Download: Extract Comments by Author from yt-dlp JSON.py


SD (E:) > yt-dlp Demonstration
Search yt-dlp Demonstration
The Real Hog Wash [iZjGb18kd7g].info
Extract Comments by Author from yt-dlp JSON
The Real Hog Wash [iZjGb18kd7g]
Glory Hole Repair [623AC6a6org]
The Real Hog Wash [iZjGb18kd7g].info
Glory Hole Repair [623AC6a6org].info
The Real Hog Wash [iZjGb18kd7g]
Glory Hole Repair [623AC6a6org]
The Real Hog W...
File Edit Format View Help
IDLE Shell 3.12.0
File Edit Shell Debug Options Window Help

```
Anyone have the name of the
music? I just call it the
critikal comercial music.


The hog wash walked so that
scrub daddy could run
```

```
Python 3.12.0 (tags/v3.12.0:0fb18b0, Oct  2 2023, 13:03:39) [MSC v.1935 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: E:\yt-dlp Demonstration\Extract Comments by Author from yt-dlp JSON.p
y
Enter the directory to search in: E:\yt-dlp Demonstration
Enter the username of the author to search for: @mikek1686
Processing file: Glory Hole Repair [623AC6a6org].info.json
No comments found by '@mikek1686' in Glory Hole Repair [623AC6a6org].info.json.
Processing file: The Real Hog Wash [iZjGb18kd7g].info.json
Found 2 comments by '@mikek1686'. Writing to file.
Processing complete.
```

This section of the code controls what goes in between each comment match. Customize to your liking.



```
new_file.write('\n\n\n'.join(matching_comments))
```

Unrelated Scripts

Here is a script which can easily calculate new calories and macros for a product, for a different serving size. You could do this with AI, but this is quicker.

Just type in the name, serving size (g), calories, fat, carbs, protein, and new serving size.

Download

Serving Size Calculator MkII.py

It could probably be turned into a GUI using python.

Here is a script which can easily calculate the midpoint between two (or multiple) hex color values. I haven't found a website which can do multiple at a time, and I've had a generally hard time getting website versions to work so I always use this.

Download

hexmerge.py

Here is a script which can quickly convert a word to a unicode lookalike. This can be done on other websites, but this can make it quick as well.

Test word → Te□t word

Download

Convert to Text to Homoglyph (Unicode Lookalikes).py

Updated 7/11/25