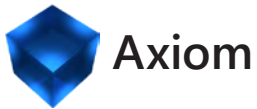


# Axiom

## An FFmpeg Interface for Windows



Axiom UI generates command scripts to be interpreted and processed by multimedia encoder, FFmpeg, and streams analyzer, FFprobe.

Convert, Cut, Resize multimedia with Lossless, Constant, and Variable Quality Modes.

It is portable and can be run from any location on the computer.

- Extract Axiom.FFmpeg.7z to a location of your choice.
- Use 7-Zip if you need a way to extract. <https://www.7-zip.org>
- Run the program Axiom.exe or create a shortcut on the Desktop.
- It will automatically detect ffmpeg.exe and ffprobe.exe in the included ffmpeg folder.
- If you move the ffmpeg folder, set Windows Environment Variables or choose path in the Settings Tab.

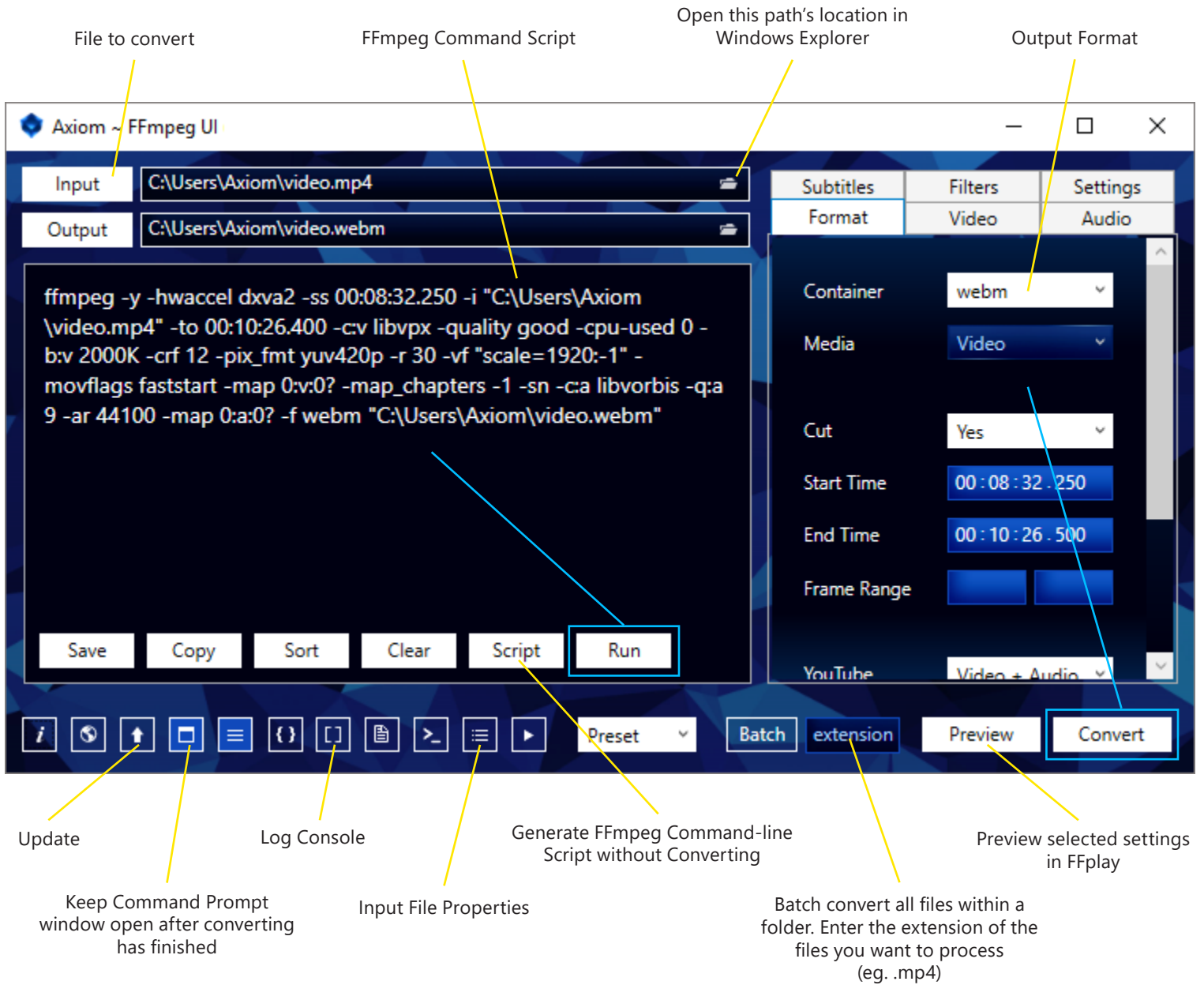


FFmpeg is separate cross-platform command-line software to record, convert and stream audio and video. It is developed and maintained at [www.ffmpeg.org](http://www.ffmpeg.org).

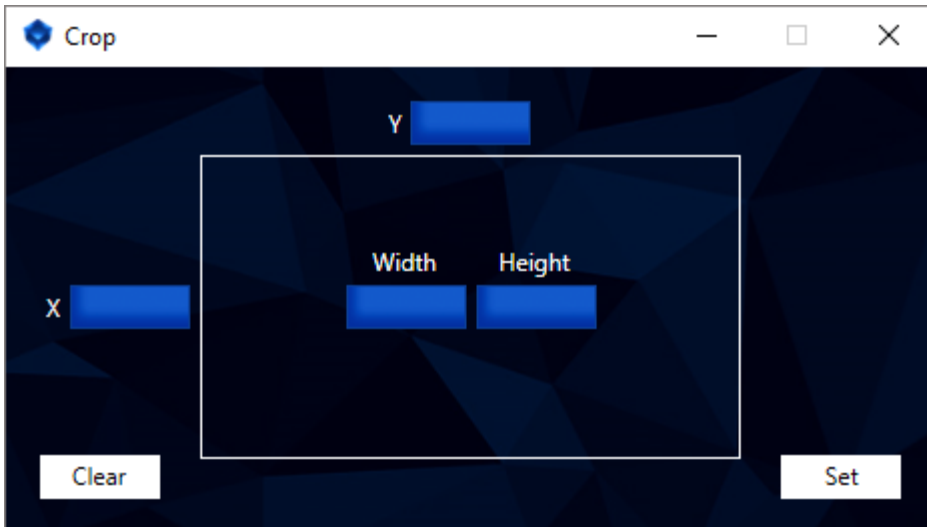
### **Enable FFmpeg Through Command Prompt** (optional)

- Move FFmpeg folder to a location of your choice, such as C:\Program Files\.
- Control Panel → System and Security → System → Advanced system settings
- Advanced Tab → Environment Variables → System variables → Path
- Add C:\Program Files\FFmpeg\bin\
- Separate multiple paths with semicolon ;
- Typing ffmpeg in Command Prompt will now execute without needing to specify a direct path.

# Interface



# Windows



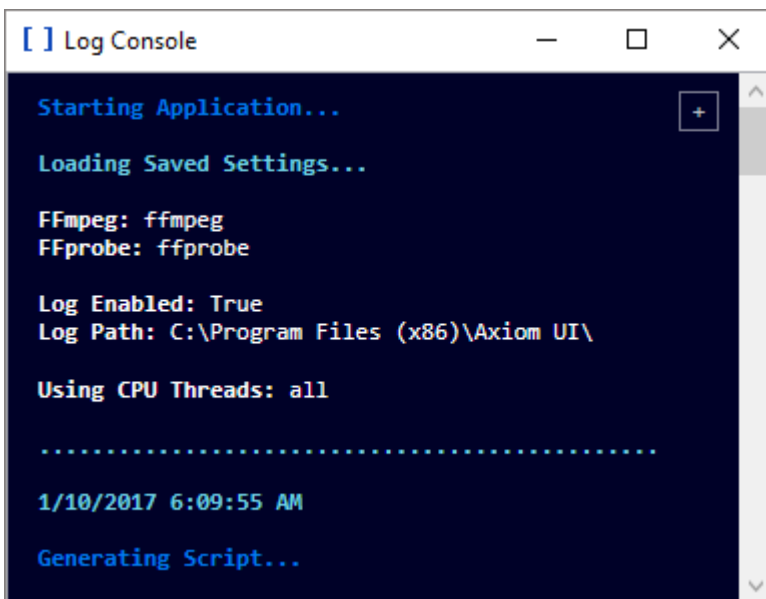
## Crop

Set the Crop Position and Aspect.

X is the position from the left.

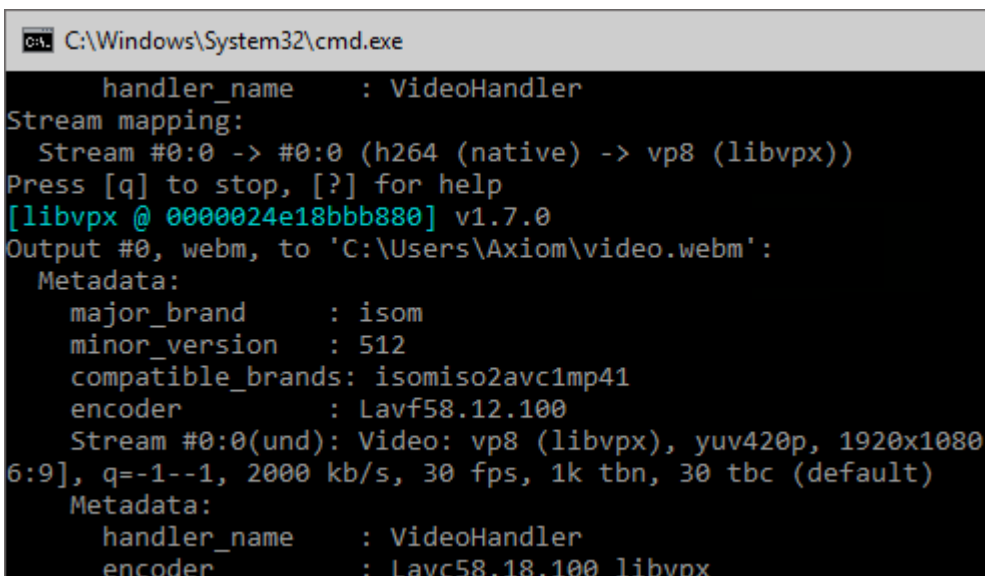
Y is the position from the top.

Width & Height is the size of the cropped area.



## Log Console

This window displays all actions performed by the program. These same actions can be saved to an output.log file in the Settings Tab.



## FFmpeg

This Command Prompt window pops up after pressing the Convert or Run button.

It displays the FFmpeg output as it processes the files.

# Interface Settings

## FFmpeg

The interface looks for ffmpeg.exe & ffprobe.exe in {Current Folder}\ffmpeg\bin\.

If you already have FFmpeg installed on your system, the FFmpeg and FFprobe included with Axiom are not needed. Custom compiled versions of FFmpeg may not work if they are missing certain codecs.

## Browse / Output

Select a file to convert, then choose a directory to output.

If you leave Output empty, the output directory will default to the same as the input directory.

## Batch

First click the Batch toggle button, then Browse for a folder to batch process the files within.

## Auto

Attempts to match the original bit-rate using FFprobe and 2-Pass encoding.

## Copy

If Auto mode selected and input/output formats are the same, Copy will automatically activate in the Codec dropdown.

## Lossless

Sets the format's correct lossless codec options or uses -crf 0, depending on format.

## Video

### Custom CRF Quality:

0 <----- 18 <----- 23 -----> 28 -----> 51  
lossless better default worse worst

### VP8/VP9:

Ultra: Bit-rate 4M + CRF 10  
High: Bit-rate 2M + CRF 12  
Med: Bit-rate 1.3M + CRF 16  
Low: Bit-rate 600K + CRF 20  
Sub: Bit-rate 250K + CRF 25

### x264:

Ultra: CRF 16  
High: CRF 20  
Med: CRF 28  
Low: CRF 37  
Sub: CRF 45

### x265:

Ultra: CRF 18  
High: CRF 21  
Med: CRF 26  
Low: CRF 35  
Sub: CRF 42

### Theora:

Ultra: qscale 10  
High: qscale 8  
Med: qscale 6  
Low: qscale 4  
Sub: qscale 2

**Video Bit-rate** is measured in bytes, K or M. (e.g 3000000 or 3000K or 3M)

## Audio

Default is Constant Bit-rate (CBR).

VBR enables a Variable Bit-rate equivalent to the CBR value selected in the Audio drop down list.

e.g. MP3: 320k (CBR) = V0 (VBR)

## Track

If your video has multiple audio tracks, such as English or Japanese, you can select which one to use or keep all.

View the video properties through a media player to find the number of the track.

The default (auto) chooses the appropriate tracks according to the output format.

## Resize

Specify a custom Width and Height.

Typing "auto" (without quotes) in a text field will select the correct width or height to match.

## Cut

Format must be 00:00:00.000

Hours:Minutes:Seconds.milliseconds (note the period instead of colon).

# FFmpeg Command-line

The screenshot shows the FFmpeg UI application window. The title bar reads "Axiom ~ FFmpeg UI". The interface is divided into several sections:

- Input/Output:** Input file is "C:\Users\Axiom\video.mp4" and Output file is "C:\Users\Axiom\video.webm".
- Command Line:** A large text area containing the following command:

```
ffmpeg -y  
-ss 00:08:32.250  
-i "C:\Users\Axiom\video.mp4"  
-to 00:10:26.500  
-c:v libvpx  
-quality good -cpu-used 0 -b:v 2000K -crf 12  
-pix_fmt yuv420p  
-r 30  
-vf "scale=1920:-1"  
-movflags faststart  
-map 0:v:0? -map_chapters -1  
-sn  
  
-c:a libvorbis  
-q:a 9  
-map 0:a:0?  
  
-map_metadata 0  
  
-f webm  
  
-threads 0  
  
"C:\Users\Axiom\video.webm"
```
- Settings Panel:** A sidebar on the right with tabs for Subtitles, Filters, and Settings. The Settings tab is active, showing various options:
  - Video Codec: VP8
  - Encode Speed: Medium
  - HW Accel: off
  - Quality: High
  - Pass: CRF
  - CRF: 12 (with a slider)
  - Bitrate: 2000K (with a VBR button)
  - Minrate: (empty)
  - Maxrate: (empty)
  - Bufsize: (empty)
  - Pixel Format: yuv420p
  - Frame Rate: 30
  - Speed: auto
  - Optimize: Web
  - Tune: none
  - Profile: none
- Buttons:** At the bottom of the command line area are buttons for Save, Copy, Sort, Clear, Script, and Run.
- Footer:** A row of icons and buttons including Preset, Batch, extension, Preview, and Convert.

```
ffmpeg -y -ss 00:08:32.250 -i "C:\Users\Axiom\video.mp4" -to 00:10:26.500 -c:v libvpx -quality  
good -cpu-used 0 -b:v 2000K -crf 12 -pix_fmt yuv420p -r 30 -vf "scale=1920:-1" -movflags fast-  
start -map 0:v:0? -map_chapters -1 -sn -c:a libvorbis -q:a 9 -map 0:a:0? -map_metadata 0 -f webm  
-threads 0 "C:\Users\Axiom\video.webm"
```

# FFmpeg Batch Script

## Change to Directory

```
cd "C:\Users\Videos\"
```

## Batch Process Filetype

```
&& for %f in (*.mp4) do (echo)
```

## FFprobe File's Size

```
& for /F "delims=" %S in ('@ffprobe -v error -select_streams v:0 -show_entries format=size -of default=noprint_ wrappers^=1:nokey^=1 "%~f" 2^>^&1') do (SET size=%S)
```

## Set %S to %size%

```
& for /F %S in ('echo %size%') do (echo)
```

## FFprobe File's Duration

```
& for /F "delims=" %D in ('@ffprobe -v error -select_streams v:0 -show_entries format=duration -of default=noprint_ wrappers^=1:nokey^=1 "%~f" 2^>^&1') do (SET duration=%D)
```

## Remove Duration Decimals for DOS

```
& for /F "tokens=1 delims=" %R in ('echo %duration%') do (SET duration=%R)
```

## Set %D to %duration%

```
& for /F %D in ('echo %duration%') do (echo)
```

## FFprobe File's Video Bit-rate

```
& for /F "delims=" %V in ('@ffprobe -v error -select_streams v:0 -show_entries stream=bit_rate -of default=noprint_ wrappers^=1:nokey^=1 "%~f" 2^>^&1') do (SET vBitrate=%V)
```

## Set %V to %vBitrate%

```
& for /F %V in ('echo %vBitrate%') do (echo)
```

## If %vBitrate% = N/A, Calculate Bit-rate (((Size\*8)/1000)/Duration)\*1000

```
& (if %V EQU N/A (SET /a vBitrate=%S*8/1000/%D*1000) ELSE (echo Video Bitrate Detected))
```

## Set %V to %vBitrate%

```
& for /F %V in ('echo %vBitrate%') do (echo)
```

## FFprobe File's Audio Bit-rate

```
& for /F "delims=" %A in ('@ffprobe -v error -select_streams a:0 -show_entries stream=bit_rate -of default=noprint_ wrappers^=1:nokey^=1 "%~f" 2^>^&1') do (SET aBitrate=%A)
```

## If Audio Variable = N/A, Default to 320k

```
& for /F %A in ('echo %aBitrate%') do (echo) & (IF %A EQU N/A (SET aBitrate=320000))
```

## Limit Audio Variable bit-rate to Output Format's maximum allowed bit-rate

```
& for /F %A in ('echo %aBitrate%') do (echo) & (IF %A gtr 500000 (SET aBitrate=500000) ELSE (echo Bitrate within Vorbis Limit of 500k))
```

## Set %A to %aBitrate%

```
& for /F %A in ('echo %aBitrate%') do (echo)
```

## Start FFmpeg

```
&& ffmpeg -y -i "C:\Users\Videos\%~f" -vcodec libvpx -quality good -cpu-used 0 -b:v %V -pass 1 -acodec libvorbis -b:a %A -map 0:v:0? -map 0:a:0? -sn -map_metadata 0 -threads 8 "C:\Users\Videos\%~nf.webm"
```

## 2-Pass Encoding

```
&& ffmpeg -y -i "C:\Users\Videos\%~f" -vcodec libvpx -quality good -cpu-used 0 -b:v %V -pass 2 -acodec libvorbis -b:a %A -map 0:v:0? -map 0:a:0? -sn -map_metadata 0 -threads 8 "C:\Users\Videos\%~nf.webm"
```

# YouTube Download

youtube-dl.exe is included in Axiom.FFmpeg.7z. Axiom will auto-detect it in its included folder.  
You can also add it to Windows Environment Variables or specify a path in the Axiom Settings Tab.

1. Paste a YouTube URL into the Input TextBox
2. Download the file only:
  - Select Preset: YouTube-DL → Video or Music, Press Download
3. Download and Convert:
  - Select any Presets or Settings you need and Press Convert
4. Generate a Script without Downloading:
  - Select any Settings you need and Press Script

## Script

### Change to Downloads Directory

```
cd /d "C:\Users\Example\Downloads\"
```

### Get Video Title from URL

```
&&
for /f "delims=" %f in (
    @"C:\Path\To\youtube-dl.exe"
    --get-filename -o "%(title)s" "https://www.youtube.com/watch?v=123456"
)
```

### Download Video as mkv

```
do (
    @"C:\Path\To\youtube-dl.exe"
    -f bestvideo[ext=mp4]+bestaudio[ext=m4a]/bestvideo+bestaudio
    "https://www.youtube.com/watch?v=123456"
    -o "C:\Users\Example\Downloads\%f.mkv"
    --merge-output-format mkv
)
```

### Convert to webm with FFmpeg

```
&&
ffmpeg -y

-i "C:\Users\Example\Downloads\%f.mkv"

-c:v libvpx
-quality good -cpu-used 0 -b:v 1300K -crf 16
-pix_fmt yuv420p
-movflags faststart
-map 0:v:0? -map_chapters -1
-sn

-c:a libvorbis
-q:a 6 -ar 44100 -rematrix_maxval 1.0 -ac 2
-map 0:a:0?

-map_metadata 0

-f webm

"C:\Users\Example\Downloads\%f.webm"
)
```