stream2clip user manual

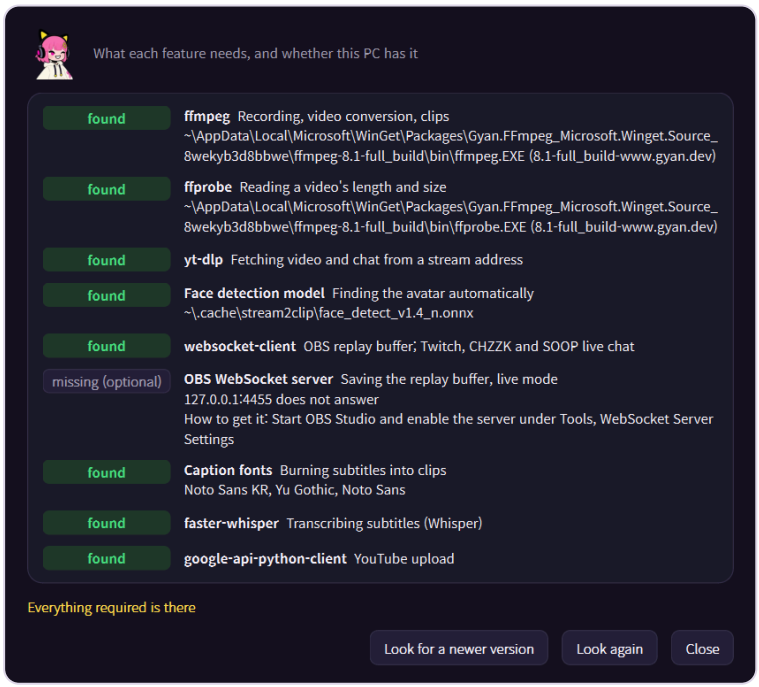
Version	v0.19
Audience	Users of the Windows installer, the executables and the Python package
Source	github.com/BudongJW/stream2clip

1. Installation

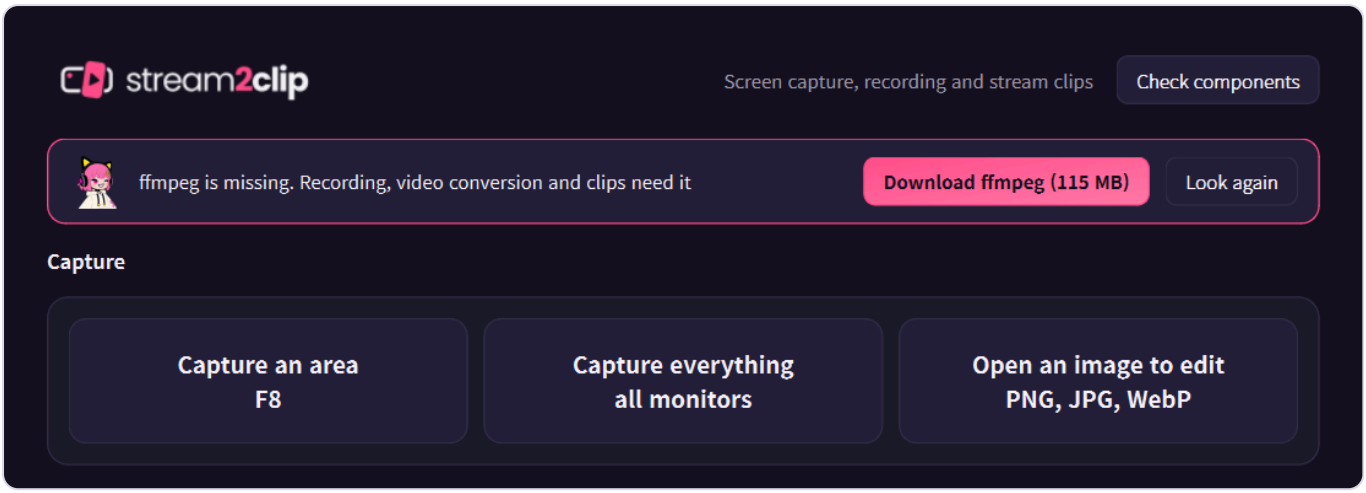
1.1 Windows installer

1. Download the Windows installer (`setup.exe`) from the releases page and run it
2. Choose the language on the first page: Korean, Japanese or English. It is used for the installer and for the app
3. Confirm the folder. No administrator rights needed
4. On the additional tasks page, keep **Download ffmpeg** ticked and install. ffmpeg (115 MB), which video conversion needs, is downloaded during the installation. The task is not shown on a PC that already has ffmpeg
5. Start stream2clip from the Start menu

- Download unticked or failed: the **Download ffmpeg** button at the top of the app's start window does it again
- Checking components: **Check components** at the top right of the start window lists what each feature needs, whether it is there and how to get it. On the command line: `stream2clip setup`



Components window. ffmpeg and the face detection model can be downloaded from it



Start window on a PC without ffmpeg. The notice goes away once ffmpeg is there

- Without installing: download the zip file, unzip it to any folder and run `stream2clip-gui.exe`. The **Download ffmpeg** button on the start window gets ffmpeg
- The language of the app can be changed later at the bottom of the launcher
- Removal: uninstall stream2clip in the installed apps list of the Windows settings. The downloaded ffmpeg is removed with it

> **Note**
>
> - Not code signed. Windows SmartScreen may show a warning
> - Smart App Control may block the programs. Use the Python package in that case
> - File integrity: compare the SHA-256 value on the releases page with the result of the command below
>
> ```
> Get-FileHash .\stream2clip-*-windows-x64.zip -Algorithm SHA256
> ```

1.2 Python package

- Requirements: Python 3.10 or later, ffmpeg
- Systems: Windows, macOS, Linux

```
pip install "stream2clip[gui,ingest,obs] @ git+https://github.com/BudongJW/stream2clip"
```

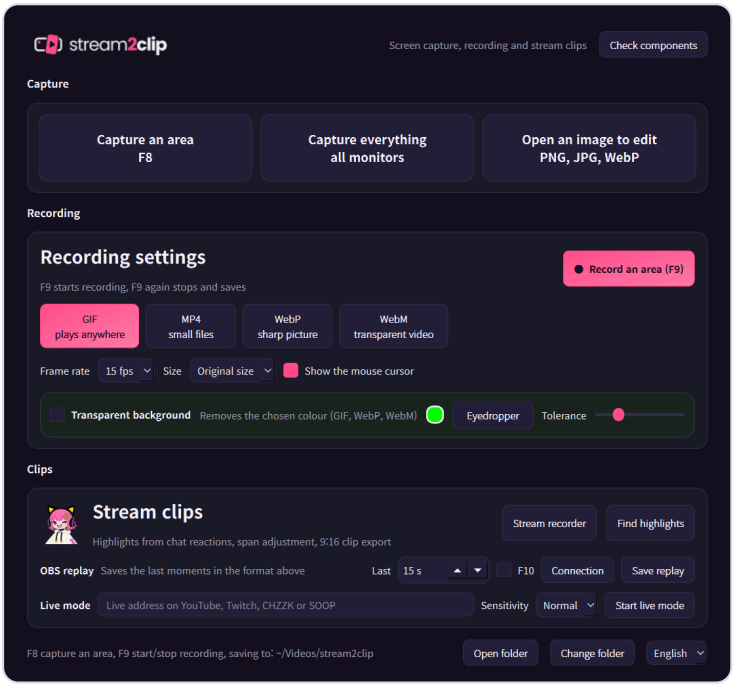
```
stream2clip gui
```

Extra	Function
`asr`	Speech-to-text subtitles (Whisper)

Extra	Function
`detect`	Automatic detection of the avatar position
`upload`	YouTube upload

Extras are added inside the square brackets of the install command. Speech-to-text subtitles and YouTube upload are provided by the Python package.

2. Basic use



Launcher. From the top: capture, screen recording, clips

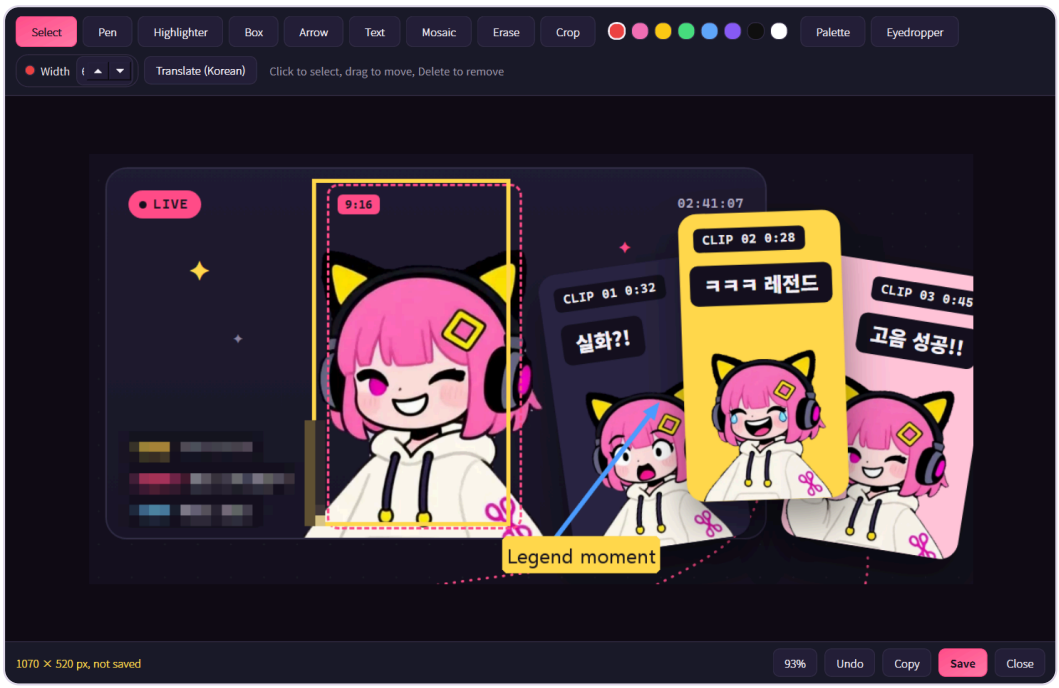
Shortcut	Function
`F8`	Capture a screen area and open it in the editor
`F9`	Select an area and start recording. Press again to stop and save
`F10`	Save the OBS replay buffer and convert it to a clip. Only when switched on in the launcher

- Shortcuts work while other programs are in use
- Default save location: `Videos\stream2clip`. Changed with **Change folder** at the bottom of the launcher
- Interface languages: Korean, Japanese, English. A change applies after a restart

- Settings are saved automatically and restored at the next start. Passwords are not saved

3. Screen capture and editing

- `F8`: freezes the screen and sends the dragged area to the editor
- **Capture everything**: capture of all monitors
- **Open an image to edit**: editing of PNG, JPG and WebP files



Editor

Key	Tool or action
`V` `P` `H`	Select, pen, highlighter
`R` `A` `T`	Box, arrow, text
`M` `E` `C`	Mosaic, erase, crop
`I`	Eyedropper (pick a colour from the screen)
`[` `]`	Decrease, increase the width
`Ctrl` + `Z`, `Ctrl` + `Y`	Undo, redo
`Ctrl` + `C`, `Ctrl` + `S`	Copy, save

Key	Tool or action
`Delete`	Remove the selected annotation

- Erase tool: dragging removes the area, clicking makes the clicked colour transparent in the whole image
- Crop tool: marks the area to keep while the original stays intact

4. Screen recording

1. Choose the output format, frame rate and size in the launcher
2. Press `F9` and drag the area to record. Recording starts at once
3. Press `F9` again to stop and save

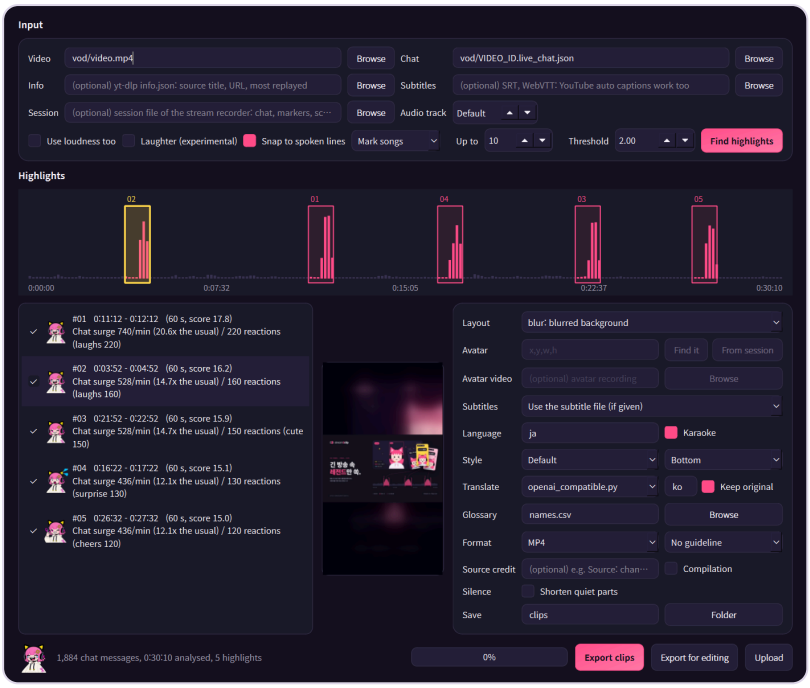
Format	Characteristics
GIF	Plays almost everywhere. Transparency has two levels
MP4	Small files
WebP	Good picture quality. Soft transparent edges
WebM	Video with transparency

4.1 Transparent background

- Output of GIF, WebP or WebM with a chosen colour (a green screen, for example) removed
- **Eyedropper** picks the colour to remove, **Tolerance** sets how similar a colour has to be

5. Highlight clips

Detection, adjustment of spans and export are done in **Find highlights** in the launcher.



Highlights window. Pink bars are detected spans, the yellow frame is the selected span

5.1 Input

Field	Content
Video	Archive file or stream address
Chat	Chat replay file. Collected automatically for a stream address
Info	Optional. Title and address of the source stream, used for the description text
Subtitles	Optional. SRT or WebVTT file

5.2 Start from a stream address

- A stream address in the Video field collects the chat replay and the stream information automatically
- Only the selected spans of the video are downloaded at export
- Collected data is kept per stream in the `sources` folder of the save folder and reused at the next run
- The preview appears once the span has been downloaded
- Loudness, laughter and singing analysis are available for a video file
- Supported: YouTube, CHZZK and SOOP archives

5.3 Procedure

1. Run **Find highlights**. Spans with chat activity above normal appear on the timeline and in the list

2. Adjust spans on the timeline. Drag the middle to move, drag an edge to change the start or end, use the wheel or a double click to zoom
3. Select the spans to export and choose the layout and format
4. Run **Export clips**. The save folder receives the clips, subtitle files (`.ass`) and description texts (`.txt`)

Video, chat, info and subtitle files can be dropped onto the window.

5.4 Compilation and editing programs

Item	Description
Source credit	The text is shown small in the top left corner of each clip. Example: Source: channel name
Music, sound effect	Enter a downloaded music file, or a name in the sound library, under Music. It plays under the clip's sound and goes lower while someone speaks. The sound effect is one of five built-in ones, or Auto (matched to chat's reaction), at the peak, the start or the end. Record a track's source and license with `stream2clip sounds add` : the description credit and the check use them. MP4 and WebM only
Sound library	Library on the music row adds a downloaded file with its license, title, author, page and credit wording. The description credit and the check result are shown while typing. A chosen track becomes the music
Loudness	Bring to -14 LUFS sets the clip's sound to YouTube's reference loudness, measured once and then raised or lowered by one fixed amount (MP4, WebM)
Shorten quiet parts	Stretches of a clip that stay quiet for more than 0.7 s are cut down to a short pause. Captions move along. Judged by loudness, so check the result before posting
Compilation	When ticked, the exported clips are also joined in stream order into `compilation.mp4` , with the chapter list `compilation.txt`
Export for editing	Saves the ticked spans as a cut list `highlights.edl` , timeline markers `markers.edl` and a timestamp list `timestamps.txt` , to go on in DaVinci Resolve, Premiere Pro and others

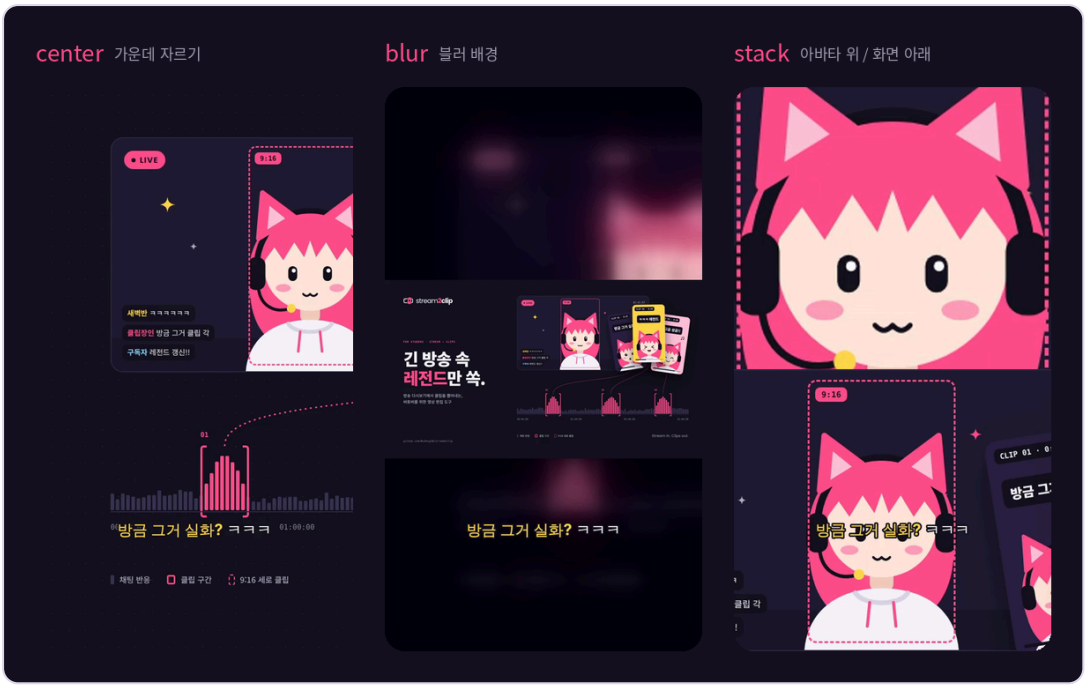
5.5 Video type and opening

Item	Description
Video type	Shorts (vertical, 15-90 s), horizontal clip (1-3 min) or horizontal compilation (about 10 minutes of 6 clips, with chapters). Sets the layout, the compilation, the context line, the caption style (sentence captions for horizontal videos, karaoke for Shorts) and the number of spans together; picked before finding highlights, it also sets their length. Most clip channels post horizontal videos
Context line	One line on the stream's topic at the top for the first 3 seconds, such as the game name in the brackets of the stream title
Peak teaser	Opens the clip with 2 seconds of its peak before the clip itself (MP4, WebM)
Thumbnail	A 1280x720 image (`clip_01.jpg`) of the sharpest frame near the peak, with the start of the title in large type. Zoomed onto the avatar when its position is set
Title, hashtags	A suggested title per clip from the line at the peak, the talent's name and the agency's name, filled into the upload window. Korean titles without an agency read `[name] line`. Hashtags at the end of the description (Korean adds the CHZZK or SOOP platform tag and kirinuki, up to five). Exaggerated words are flagged by the check
Collab guests, channel links	Collab guests (separated by commas) go at the end of the title as `(w/ name)`. Channel links go under the original channel line of the description. The contact line adds a sentence for streamers without clip guidelines. The description is written in the title's language, with a line for the stream date
Upload category	The category in the upload window. Auto picks Gaming when the original stream was a game stream, People & Blogs otherwise. A category can also be picked by hand

Item	Description
Effects	Only at the peak: a short zoom and one emphasised caption word (Subtle), plus a tag for the chat's reaction (Variety). Left out when neither the moment the sound rises nor a VTube Studio hotkey press is clear. Reduce motion emphasises the word more instead of zooming
Reused content check	Flags clips only cut from the stream, with no captions, context line or teaser. Under YouTube's reused content policy, commentary or captions by the editor are needed before posting

5.6 Layouts

Name	Composition
blur	Stream picture in the centre, blurred copy filling the space above and below
center	Centre of the picture enlarged to the vertical frame. Left and right are cut off
fit	Whole picture scaled down to fit
stack	Avatar on top, game picture below. **Find it** detects the avatar position
pip	Game picture fills the frame, avatar in a corner. **Find it** detects the avatar position
Avatar video	Optional for stack and pip. A recording of the avatar alone is laid over the game with its green background removed. A transparent video is used as it is



Layout examples, made with the brand image as the stream picture

5.7 Detection options

- Threshold: a lower value finds more spans, a higher value keeps only strong reactions
- Up to: upper limit of the number of spans
- Songs: marks or leaves out spans that overlap singing. Clips of singing streams are often restricted for copyright reasons

5.8 Download of a whole stream

The command below saves the video, chat and stream information to the `vod` folder when the whole video is needed.

```
stream2clip.exe fetch "STREAM_ADDRESS" -o vod --video
```

6. Subtitles and translation

Item	Content
Subtitle file	SRT and WebVTT. YouTube automatic captions can be used
Karaoke	Colour follows the spoken word. Applies to subtitles with word timing

Item	Content
Style	27 looks and 7 places. Default, Box, Pink highlight and Small use an installed font that suits the language of the captions; the others use free fonts (14 Korean, 6 Japanese, 4 English, SIL Open Font License), downloaded on first use and checked against their checksums. Places: Bottom, Bottom edge, Middle, Top, and for Shorts right under, over and right above the stream picture. Sizes follow Korean clip channels (about 4.5% of the height in Shorts, 5.5% in horizontal clips)
Snap to spoken lines	With a subtitle file, the start and the end of each span move to the edge of a subtitle line. Limit 3 seconds. Applied when highlights are searched
Transcription	Subtitles made from speech when there is no file (Whisper small, medium, large-v3-turbo, large-v3). The large models need a GPU to be quick. Provided by the Python package
Translation	**Choose a translator file** in the translator list with `translators\openai_compatible.py` translates through an OpenAI compatible address such as a local Ollama
Glossary	CSV with two columns, source and translation. Fixes names and other set terms. Transcription takes the names as hints, and misheard spellings in a heard column are corrected to the names
Keep original	Original and translation shown on two lines
Caption check	For each exported clip, lines too fast to read, too wide for the frame, shown too briefly or repeated, and phrases transcription invents over silence are listed in clips.json and counted

- No translation service is built in. The user connects a translator
- Translation quality depends on the connected model

7. OBS replay

Saves the recent moments held in the OBS replay buffer and converts them to a clip.

7.1 Preparation in OBS

1. Enable the replay buffer in the Output section of the OBS settings. It is off after a new installation
2. Enable the server in the WebSocket server settings of the OBS Tools menu and check the password
3. Start the replay buffer in OBS when the stream starts

7.2 Settings and use in stream2clip

1. Enter the OBS password in **Connection** in the launcher. The password is not saved and is entered at every start
2. Set the length to keep in **Last** and run **Save replay**. The format and size of the recording settings are used
3. With **F10** checked, `F10` saves while other programs are in use. Other programs do not receive F10 meanwhile

8. Live mode

Detects a chat surge during a stream, saves the OBS replay buffer and cuts the moment out as a clip of about 35 seconds.

8.1 Prerequisites

- OBS replay buffer running and the WebSocket connection set up
- Maximum replay time in OBS of 60 seconds or more. The default of 20 seconds loses the beginning of the moment

8.2 Procedure

1. Enter a live address on YouTube, Twitch, CHZZK or SOOP in the Live mode field of the launcher
2. Choose the sensitivity. High saves on small surges, Low only on large ones
3. Run **Start live mode**. A chat surge creates a clip named `live_DATE_TIME` in the save folder
4. End with **Stop live mode**

8.3 How it decides

- Chat volume, number of chatters, reaction words and paid messages per 5 seconds are compared with the 15 minutes before
- The first 2 minutes only collect the baseline. Clips are at least 90 seconds apart
- The moment is taken to be 8 seconds before the surge. The clip keeps 20 seconds before and 10 seconds after
- `!clip` typed in chat by the streamer or a moderator saves a clip at once, surge or not
- Supported: live YouTube streams, Twitch, CHZZK, SOOP. Adult-only and password streams on CHZZK and SOOP are excluded

8.4 Command line

```
stream2clip.exe live "https://www.youtube.com/watch?v=VIDEO_ID" -o clips
```

Option	Function
`--threshold`	Level that triggers a save. Default 3.5. A higher value saves less often
`--no-obs`	Reports surges without saving
`--simulate`	Shows when a finished stream would have triggered, from its chat file
`--delay`	Reaction delay of viewers. Streamers add their stream latency

9. Stream recorder

Records the chat, markers, OBS scenes and the position and expression of the VTube Studio avatar into a session file during the stream, for making clips afterwards. OBS records the video.

9.1 Procedure

1. Open **Stream recorder** in the clip area of the start window
2. Enter the live address of the streamer's own channel and choose whether to use OBS and VTube Studio. In VTube Studio, switch the API on in the settings and press Allow in the plugin popup the first time
3. Enter the name of the OBS source that shows VTube Studio. It places the avatar box when clips are made
4. Run **Start recording**. At a moment to keep, press **Mark now** or F11 (when switched on)
5. After the stream, **Stop recording**, then **Find highlights** opens the session file in the highlights window

9.2 What is recorded and what for

- Chat and markers: highlights are found without downloading the chat replay. Marked moments always become clips
- OBS scenes: highlights in chosen scenes such as a waiting or ending screen are left out
- VTube Studio outline and its place in OBS: the avatar box of vertical clips
- Second 0 of the video is when OBS started streaming, or recording when it did not stream
- Viewer votes: 3 viewers typing `!clip` within 30 seconds leave a marker
- OBS widget: `http://127.0.0.1:8790/widget` added as an OBS browser source shows the votes and markers on stream
- Chatters are recorded only as values that do not identify them, and the file stays on this PC

9.3 Settings in the highlights window

- Put the session file in the **Session** box. A session file put in the chat box or dropped on the window moves there too. With the chat box empty, highlights are found from the session's chat and markers
- **Second 0**: choose OBS stream start for a platform's replay and OBS recording start for an OBS recording file. When the video starts later than that, enter the difference in seconds
- **Skip scenes**: parts of scene names separated by commas. Highlights with half or more of their length in such a scene are left out
- **Avatar motion (experimental)**: adds to the score of moments in which the avatar moved a lot
- **From session** in the avatar row of the export settings: fills the avatar box from the outline during the ticked highlights
- **Audio track**: the track number when OBS recorded the microphone on a track of its own. Used for subtitles, silences and loudness, and the sound of the clips does not change

9.4 Command line

```
stream2clip.exe session record --chat "https://chzzk.naver.com/live/CHANNEL_ID" --vts --
vts-source "VTube Studio"
```

```
stream2clip.exe auto vod.mp4 -o clips --session session.jsonl --skip-scene waiting --
layout stack --avatar session
```

10. Command line

Every function of the app is available through `stream2clip.exe`. Used for processing several streams in one go.

Command	Function
`fetch`	Download of stream information, chat, video and subtitles from a stream address
`auto`	From a video with chat, or from a stream address, to finished clips
`highlights`, `clips`	Detection and export as separate steps
`convert`	Cuts a span and area of a video and converts it to GIF, WebP or MP4
`record`, `capture`	Recording and capture of a screen area
`replay`, `live`	OBS replay save, live mode

Command	Function
`check`	Compares a description text with the guideline items of an agency

```
stream2clip.exe auto "STREAM_ADDRESS" --top 5 -o clips
```

```
stream2clip.exe auto vod\video.mp4 --chat vod\VIDEO_ID.live_chat.json --info
vod\info.json -o clips
```

```
stream2clip.exe convert vod.mp4 --start 1:02:03 --end 1:02:18 --width 480 -o
highlight.webp
```

All options: add `--help` to a command. More examples in the README (Korean).

11. Troubleshooting

Symptom	Action
Program blocked	Caused by the missing code signature. Use the Python package where Smart App Control is on
Newer version notice	Shown at the top of the start window. **Open the download page** gets the new installer, **Stop asking** ends the check. It asks at most once a day and sends nothing about the user
ffmpeg not found	Use the **Download ffmpeg** button on the app's start window, or `stream2clip setup --ffmpeg` on the command line. After installing it yourself: **Look again**, or restart the program
No spans found	Lower the threshold and run again. Check that the chat file belongs to the stream
YouTube asks for sign-in	Pass browser cookies on the command line with `--cookies-from-browser chrome`
Replay buffer not running	Enable the replay buffer in the Output section of the OBS settings and start it. OBS does not start the buffer for a scene without sources

Symptom	Action
No connection to OBS	Check that the WebSocket server is enabled, and the password and port (default 4455)
Live mode clip misses its beginning	Set the maximum replay time in OBS to 60 seconds or more
F8 or F9 without effect	Close the other program that uses the same key and restart

Other problems: GitHub issues.

12. Copyright and guidelines



- A clip is a derivative work of the source stream. The guidelines of the creator and of the agency apply
- Choosing an agency in the guideline list of the export settings writes a description text with the source title and address and checks the guideline items
- Items checked: members-only stream, public archive, channel registration, disclosure of AI use
- The results are for reference and are not legal advice. Read the original guidelines of the agency and the creator before uploading

Stream in. Clips out.