stream2clip 取扱説明書

対象バージョン	v0.19
対象	Windows用インストーラー、実行ファイルおよびPythonパッケージの利用者
配布元	github.com/BudongJW/stream2clip

---

1. インストール

1.1 Windows用インストーラー

1. リリースページからWindows用インストーラー(`setup.exe`)をダウンロードして実行
2. 最初の画面で言語を選択: 韓国語、日本語、英語。
   選択した言語がインストーラーとアプリの表示言語として適用
3. インストール先を確認。管理者権限は不要
4. 追加タスクの画面で **ffmpeg をダウンロード** を選択したままインストール。
   動画変換に必要なffmpeg(115MB)をインストール中にダウンロード。
   ffmpegが既にあるPCではこの項目は表示されない
5. スタートメニューからstream2clipを起動

- ダウンロードを外した場合や失敗した場合: アプリの開始画面上部の **ffmpeg をダウンロード** ボタンで再実行
- コンポーネントの確認: 開始画面右上の **コンポーネント確認** で、
  機能ごとに必要なコンポーネントの有無と対処方法を表示。コマンドラインでは `stream2clip setup`



コンポーネント確認ウィンドウ。ffmpeg と顔検出モデルはこのウィンドウからダウンロード可能



ffmpegがないPCの開始画面。ffmpegの準備ができると案内は消える

- インストールせずに使用: zipファイルをダウンロードして任意のフォルダに解凍し、`stream2clip-gui.exe` を実行。ffmpegは開始画面の **ffmpeg をダウンロード** ボタンで準備
- アプリの表示言語はインストール後もランチャー下部で変更可能
- アンインストール: Windowsの設定のインストール済みアプリ一覧からstream2clipを削除。
  ダウンロードしたffmpegも一緒に削除

注記

- コード署名は未適用。Windows SmartScreenの警告が表示される場合あり
- Smart App Controlが有効な環境では実行がブロックされる場合あり。
  その場合はPythonパッケージを使用
- ファイルの整合性は、リリースページのSHA-256値と下記コマンドの結果を照合して確認

```
Get-FileHash .\stream2clip-*-windows-x64.zip -Algorithm SHA256
```

1.2 Pythonパッケージ

- 要件: Python 3.10以上、ffmpeg
- 対応OS: Windows、macOS、Linux

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

```
stream2clip gui
```

追加項目	機能
`asr`	字幕の文字起こし(Whisper)
`detect`	アバター位置の自動検出
`upload`	YouTubeアップロード

追加項目はインストールコマンドの角括弧内に記述。
字幕の文字起こしとYouTubeアップロードはPythonパッケージで提供。

2. 基本的な使い方



ランチャー画面。上からキャプチャ、画面録画、クリップ作成

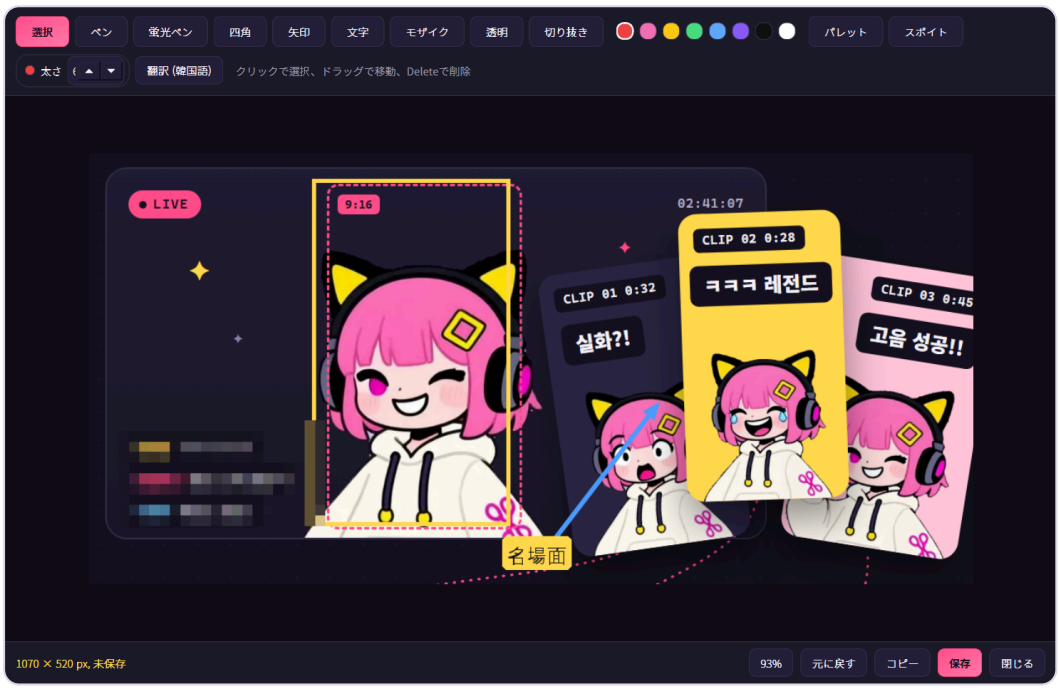
ショートカット	機能
`F8`	画面の範囲をキャプチャして編集画面を起動
`F9`	範囲を指定して録画開始。もう一度押すと停止して保存
`F10`	OBSリプレイバッファを保存してクリップに変換。 ランチャーで有効にした場合のみ

- ショートカットは他のプログラムの使用中も動作
- 保存先の初期値は `ビデオ\stream2clip`。ランチャー下部の**フォルダを変更**で変更

- 表示言語は韓国語、日本語、英語に対応。変更は再起動後に適用
- 設定は自動保存され、次回起動時に復元。パスワードは保存しない

3. 画面キャプチャと編集

- `F8`: 画面を静止し、ドラッグした範囲を編集画面へ送る
- **全画面キャプチャ**: すべてのモニターをキャプチャ
- **画像を開いて編集**: PNG、JPG、WebPファイルの編集



編集画面

キー	ツールと操作
`V` `P` `H`	選択、ペン、蛍光ペン
`R` `A` `T`	四角、矢印、文字
`M` `E` `C`	モザイク、透明、切り抜き
`I`	スポイト(画面から色を取得)
`[` `]`	太さを減らす、増やす
`Ctrl` + `Z`, `Ctrl` + `Y`	元に戻す、やり直す
`Ctrl` + `C`, `Ctrl` + `S`	コピー、保存

キー	ツールと操作
`Delete`	選択した注釈を削除

- 透明ツール: ドラッグで範囲を消去、クリックでその色を画像全体から透明化
- 切り抜きツール: 元画像を保持したまま、残す範囲のみを指定

4. 画面録画

1. ランチャーで出力形式、フレームレート、サイズを選択
2. `F9` を押して録画範囲をドラッグ。直ちに録画開始
3. `F9` をもう一度押すと停止して保存

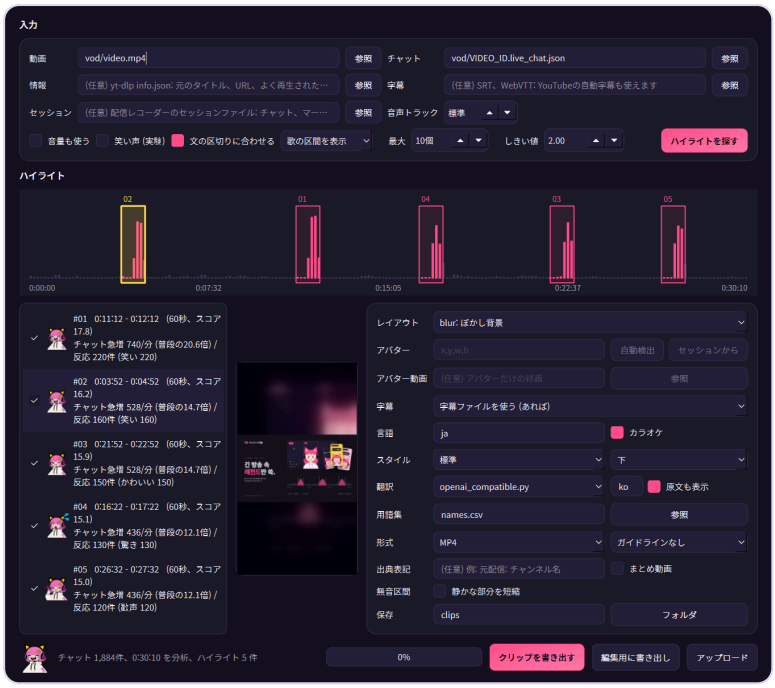
形式	特徴
GIF	再生互換性が高い。透過は2段階
MP4	ファイルサイズが小さい
WebP	画質が良い。滑らかな透過の縁に対応
WebM	透過動画に対応

4.1 透過背景で録画

- 指定した色(グリーンバックなど)を除去したGIF、WebP、WebMを出力
- **スポイト**で除去する色を指定、**許容範囲**で類似色の除去範囲を調整

5. ハイライトクリップの作成

ランチャーの**ハイライトを探す**で検出、区間の調整、書き出しを行う。



ハイライト画面。ピンクの棒は検出区間、黄色の枠は選択中の区間

5.1 入力

項目	内容
動画	アーカイブのファイルまたは配信URL
チャット	チャットリプレイのファイル。配信URLの入力時は自動収集
情報	任意。元配信のタイトルとURLを概要欄の作成に使用
字幕	任意。SRT、WebVTTファイル

5.2 配信URLから開始

- 動画欄に配信URLを入力すると、チャットリプレイと配信情報を自動収集
- 書き出し時に、選択した区間の映像のみをダウンロード
- 収集したデータは保存フォルダの `sources` フォルダに配信ごとに保管し、再実行時に再利用
- プレビューは該当区間のダウンロード完了後に表示
- 音量、笑い声、歌の区間の分析は動画ファイルの入力時に使用可能
- 対応: YouTube、CHZZK、SOOPのアーカイブ

5.3 作業手順

1. **ハイライトを探す**を実行。チャット反応が通常より高い区間をタイムラインと一覧に表示

2. タイムラインで区間を調整。中央のドラッグで移動、端のドラッグで開始と終了を調整、ホイールまたはダブルクリックで拡大
3. 書き出す区間を選択し、レイアウトと形式を指定
4. **クリップを書き出す**を実行。保存フォルダにクリップ、字幕ファイル(`.ass`)、概要欄(`.txt`)を作成

動画、チャット、情報、字幕のファイルはウィンドウにドラッグ＆ドロップで入力可能。

5.4 まとめ動画と編集ソフト連携

項目	内容
出典表記	入力した文言をクリップの左上に小さく表示。例: 元配信: チャンネル名
BGM、効果音	BGM に自分でダウンロードした音源ファイル、 または音源リストの名前を入力。クリップの音の下に流れ、 話している間はさらに下がる。効果音は内蔵の5種類、または自動 (チャットの反応に合わせる) を最高潮、最初、最後に配置。 音源の出典とライセンスを `stream2clip sounds add` で記録すると概要欄のクレジットと点検に使用。MP4、WebM で使用
音源リスト	BGM の行のリストから、ダウンロードしたファイルをライセンス、タイトル、作者、URL、クレジット文言と一緒に追加。 入力中に概要欄のクレジットと点検結果を表示。選んだ曲は BGM として使用
音量	-14 LUFS に合わせるを選ぶと、クリップの音を YouTube の基準音量に合わせる。一度測定してから一定に調整 (MP4、WebM)
静かな部分を短縮	クリップ内で 0.7 秒を超えて静かな部分を短い間に短縮。字幕も一緒に移動。 音量で判定するため、投稿前に結果を確認
まとめ動画	選択すると、書き出したクリップを配信の順につないだ `compilation.mp4` とチャプター一覧 `compilation.txt` も保存
編集用に書き出し	チェックした区間をカットリスト `highlights.edl` 、タイムラインマーカー `markers.edl` 、タイムスタンプ一覧 `timestamps.txt` として保存。DaVinci Resolve、Premiere Pro などで読み込んで編集を継続

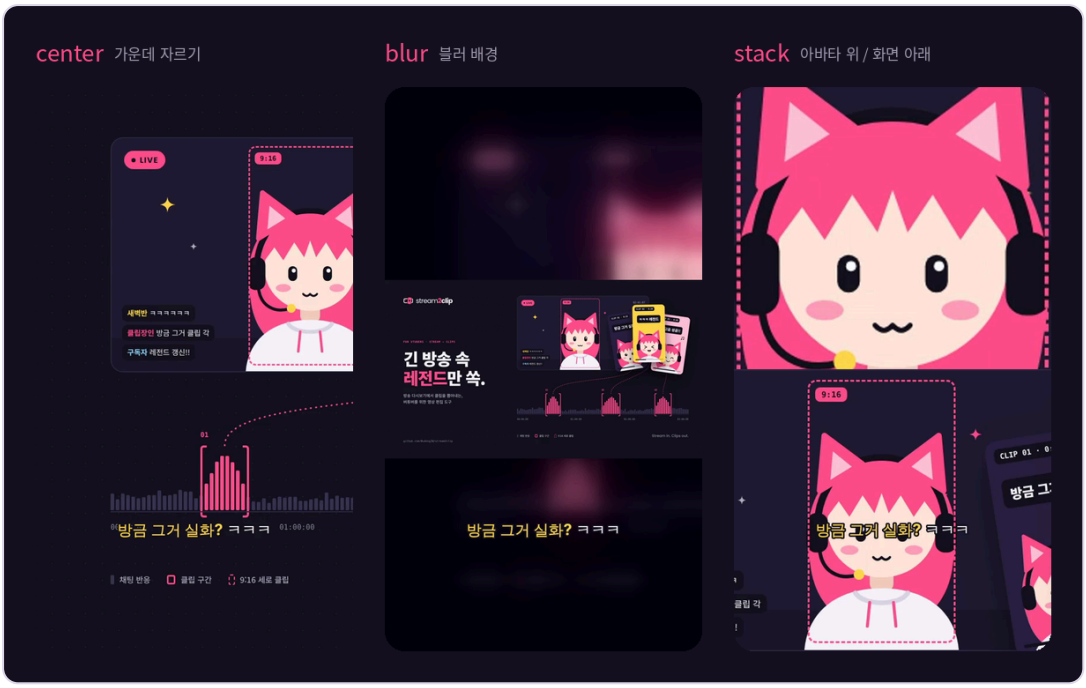
5.5 動画の種類と冒頭

項目	内容
動画の種類	Shorts (縦, 15~90秒)、横の切り抜き (1~3分)、横のまとめ (クリップ6本をつないだ10分前後の動画とチャプター) から選択。レイアウト、まとめ動画、状況説明、字幕の方式 (横は文単位の字幕、Shortsはカラオケ)、区間数をまとめて切り替え、ハイライトを探す前に選ぶと区間の長さも適用。実際の切り抜きチャンネルはほとんどが横動画
状況説明	クリップの最初の3秒、上に配信の話題を1行表示。配信タイトルの括弧の中のゲーム名などを使用
ちょい見せ	クリップの冒頭で一番盛り上がる2秒を先に見せてから本編 (MP4, WebM)
サムネイル	最高の瞬間の近くで一番鮮明な場面に、タイトルの最初の部分を大きく入れた1280x720の画像 (`clip_01.jpg`)。アバターの位置を指定した場合はアバターを中心に拡大
タイトル、ハッシュタグ	クリップごとに最高の瞬間のセリフ、タレント名、事務所名からタイトルを提案。アップロード画面のタイトルの初期値。事務所のない韓国語タイトルは `［名前］ セリフ`。ハッシュタグは概要欄の最後に追加(韓国語はCHZZK、SOOPのプラットフォームタグとキリヌキを加えて最大5つ)。誇張表現は点検で警告
コラボ相手、チャンネルリンク	コラボ相手(カンマ区切り)はタイトルの最後に `（w/名前）`。チャンネルリンクは概要欄の元チャンネルの行の下に追加。連絡文を選ぶと、ガイドラインのない配信者向けの一文を追加。概要欄はタイトルと同じ言語で書き、配信日の行を含む
アップロードのカテゴリ	アップロード画面のカテゴリ。自動は元配信がゲーム配信ならゲーム、それ以外はブログ。手動でも選択可能

項目	内容
エフェクト	最高の瞬間1か所だけに短いズームと字幕の単語の強調(控えめ)、チャットの反応タグを追加(バラエティ)。音が大きくなる瞬間やVTube Studio のホットキーを押した瞬間がはっきりしなければ入れない。 動きを減らす選択時はズームの代わりに単語をより大きく強調
再利用コンテンツの点検	字幕、状況説明、 ちょい見せなしで切り出しただけのクリップを警告。YouTubeの再利用コンテンツのポリシー上、 投稿前に編集者自身の解説や字幕の追加が必要

5.6 レイアウト

名称	構成
blur	配信画面を中央に配置し、上下の余白をぼかした背景で埋める
center	画面の中央部を縦画面に合わせて拡大。左右は切り取られる
fit	画面全体が見えるように縮小
stack	上にアバター、下にゲーム画面。**自動検出**でアバターの位置を検出
pip	ゲーム画面を全体に表示し、隅にアバターを配置。 **自動検出**でアバターの位置を検出
アバター動画	stack、pipで任意に使用。アバターだけを録画した動画を指定すると、 緑の背景を取り除いて合成。透過背景の動画はそのまま使用



レイアウトの例。ブランド画像を配信画面として使用して作成

5.7 検出オプション

- しきい値: 低いほど検出区間が増加、高いほど反応の大きい区間のみ検出
- 最大: 検出する区間数の上限
- 歌の区間: 歌と重なる区間を表示または除外。
  歌配信は著作権の都合でクリップ作成が制限される場合が多い

5.8 配信データの一括ダウンロード

動画全体が必要な場合は、下記コマンドで動画、チャット、配信情報を `vod` フォルダに保存。

```
stream2clip.exe fetch "配信URL" -o vod --video
```

6. 字幕と翻訳

項目	内容
字幕ファイル	SRT、WebVTTに対応。YouTubeの自動字幕も使用可能
カラオケ	発話中の単語に合わせて色を変更。単語ごとの時刻情報がある字幕に適用

項目	内容
スタイル	見た目27種と位置7種を選択。標準、ボックス、ピンク強調、 小さめは字幕の言語に合うインストール済みフォント、 その他は無料フォント(韓国語14種、日本語6種、英語4種、SIL Open Font License)を初回使用時にダウンロード(検証値を確認)。位置は下、下端、中央、上と、Shorts用の配信画面のすぐ下、画面上、画面のすぐ上。 サイズは韓国の切り抜きチャンネル基準(Shortsは高さの約4.5%、 横型クリップは約5.5%)
文の区切りに合わせる	字幕ファイルがある場合、区間の開始と終了を字幕行の境界へ移動。 移動の上限は3秒。ハイライト検索の実行時に適用
文字起こし	字幕がない場合に音声から字幕を作成(Whisper small、medium、large-v3-turbo、large-v3)。large系はGPU推奨。Pythonパッケージで提供
翻訳	翻訳一覧の**翻訳プログラムのファイルを選ぶ**で `translators\openai_compatible.py` を選択すると、 OpenAI互換のアドレス(ローカルのOllamaなど)で翻訳
用語集	原文と訳語の2列で構成したCSV。人名などの固定表記に使用。 文字起こしでは名前のヒントに使い、heard 列に書いた聞き間違いの表記を字幕で名前に修正
原文も表示	原文と訳文を2行で表示
字幕の確認	書き出したクリップごとに、読むには速い行、画面幅を超える行、 短すぎる表示、繰り返す行、文字起こしが無音部分で作りがちな文言を clips.json に記録し件数を表示

- 翻訳サービスは内蔵しておらず、利用者が翻訳プログラムを接続
- 翻訳の品質は接続したモデルに依存

7. OBSリプレイの保存

OBSのリプレイバッファに保持された直近の区間を保存し、クリップに変換。

7.1 OBSの事前設定

1. OBS設定の出力メニューでリプレイバッファを有効化。新規インストール時の初期値は無効
2. OBSツールメニューのWebSocketサーバー設定でサーバーを有効化し、パスワードを確認
3. 配信開始時にOBSでリプレイバッファを開始

7.2 stream2clipの設定と使用

1. ランチャーの**接続設定**にOBSのパスワードを入力。パスワードは保存しないため、起動のたびに入力
2. **最後の**で残す長さを指定し、**リプレイを保存**を実行。画面録画設定の形式とサイズで保存
3. **F10**を選択すると、他のプログラムの使用中も `F10` で保存。
   使用中は他のプログラムがF10の入力を受け取らない

8. ライブモード

配信中のチャット急増を検知してOBSリプレイバッファを保存し、
その場面を約35秒のクリップとして作成。

8.1 前提条件

- OBSリプレイバッファの実行とWebSocket接続設定の完了
- OBSの最大リプレイ時間を60秒以上に設定。初期値の20秒では場面の冒頭が欠落

8.2 使用手順

1. ランチャーのライブモード欄にYouTube、Twitch、CHZZK、SOOPのライブURLを入力
2. 感度を選択。高は小さな急増でも保存、低は大きな急増のみ保存
3. **ライブモード開始**を実行。チャット急増時に保存フォルダへ `live_日付_時刻` 形式のクリップを作成
4. **ライブモード停止**で終了

8.3 動作基準

- 5秒単位のチャット量、参加者数、反応語彙、有料メッセージを直前の15分と比較
- 開始後2分間は基準の収集のみ。クリップ間の最小間隔は90秒
- 急増の8秒前を場面の時点と判断し、前20秒と後10秒を含める
- 配信者またはモデレーターがチャットに `!clip` を入力すると、急増の有無に関係なく即時保存
- 対応: YouTubeライブ配信、Twitch、CHZZK、SOOP。CHZZK、
  SOOPの成人認証配信とパスワード配信は対象外

8.4 コマンドラインでの実行

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

オプション	機能
`--threshold`	発動基準。初期値3.5。高いほど保存回数が減少

オプション	機能
`--no-obs`	保存せずに急増の発生のみ出力
`--simulate`	終了した配信のチャットファイルで発動時点を確認
`--delay`	視聴者の反応の遅れ。配信者は配信の遅延分を加算して使用

9. 配信レコーダー

配信中のチャット、マーク、OBSのシーン、VTube
Studioのアバターの位置と表情をセッションファイルに記録し、配信後の切り抜き制作に使用。
映像はOBSが録画。

9.1 使い方

1. スタート画面のクリップ欄で**配信レコーダー**を開く
2. 自分のチャンネルの配信URLを入力し、OBSとVTube Studioの使用を選択。VTube
   Studioは設定でAPIをオンにし、初回接続時にVTube Studioに表示されるプラグイン許可画面で許可
3. OBSでVTube Studioを表示しているソース名を入力。切り抜き制作時のアバター領域の自動指定に使用
4. **記録開始**を実行。残したい場面で**今をマーク**またはF11(オン時)
5. 配信終了後に**記録停止**、**ハイライトを探す**でハイライト画面にセッションファイルを入力

9.2 記録内容と用途

- チャットとマーク: チャットリプレイを取得せずにハイライト検出。
  マークした場面は必ずクリップに含む
- OBSのシーン: 待機画面、エンディングなど指定したシーンのハイライトを除外
- VTube Studioの輪郭とOBSでの配置: 縦型クリップのアバター領域を指定
- 映像の0秒はOBSの配信開始時点。配信の記録がなければ録画開始時点
- 視聴者投票: 視聴者3人が30秒以内に`!clip`を入力するとマークとして記録
- OBSウィジェット:
  OBSのブラウザソースに`http://127.0.0.1:8790/widget`を追加すると投票状況とマークを配信画面に表示
- チャットの投稿者は識別できない値でのみ記録し、ファイルはこのPCにのみ保存

9.3 ハイライト画面の設定

- **セッション**欄にセッションファイルを入力。
  チャット欄に入れた場合や画面にドロップした場合もセッション欄に移動。
  チャット欄が空の場合はセッションのチャットとマークで検出

- **0 秒の基準**: プラットフォームのアーカイブはOBS配信開始、OBSの録画ファイルはOBS録画開始を選択。動画が基準より遅く始まる場合はその差を秒単位で入力
- **除外シーン**: シーン名の一部をカンマ区切りで入力。半分以上がそのシーンに含まれるハイライトを除外
- **アバターの動き (実験)**: アバターの動きが大きい場面に点数を追加
- 書き出し設定のアバター行の**セッションから**: チェックしたハイライト区間の輪郭でアバター領域を入力
- **音声トラック**: OBSでマイクを別トラックに録音した場合はそのトラック番号を選択。字幕認識、無音区間、音量の判定に使用し、クリップの音声は変更なし

9.4 コマンドライン

```
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 待機 --layout stack --avatar session
```

10. コマンドライン

アプリの全機能は `stream2clip.exe` で実行可能。複数の配信の一括処理に使用。

コマンド	機能
`fetch`	配信URLから配信情報、チャット、動画、字幕をダウンロード
`auto`	動画とチャット、または配信URLから、 ハイライト検出とクリップ作成までを実行
`highlights`, `clips`	検出と書き出しを個別に実行
`convert`	動画の区間と範囲を切り出してGIF、WebP、MP4に変換
`record`, `capture`	画面範囲の録画とキャプチャ
`replay`, `live`	OBSリプレイの保存、ライブモード
`check`	概要欄を事務所のガイドライン項目と照合

```
stream2clip.exe auto "配信URL" --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
```

全オプションは各コマンドに `--help` を付けて確認。その他の例はREADME(韓国語)を参照。

11. トラブルシューティング

症状	対処
プログラムの実行がブロックされる	コード署名未適用によるブロック。Smart App Controlが有効な環境ではPythonパッケージを使用
新バージョンの通知	開始画面の上部に表示。 **ダウンロードページを開く**で新しいインストーラーを取得、**通知をオフ**で確認を停止。確認は1日1回以内で、ユーザー情報は送信しない
ffmpegが見つからない	アプリの開始画面の **ffmpeg をダウンロード** ボタンを使用。コマンドラインでは `stream2clip setup --ffmpeg`。自分でインストールした場合は **再確認** ボタン、またはプログラムを再起動
区間が検出されない	しきい値を下げて再実行。 チャットファイルが該当配信のものか確認
YouTubeでログインを要求される	コマンドラインで `--cookies-from-browser chrome` オプションによりブラウザのCookieを渡す
リプレイバッファが動作していない	OBS設定の出力メニューでリプレイバッファを有効化して開始。シーンにソースがない場合、 OBSはバッファを開始しない
OBSに接続できない	WebSocketサーバーの有効化、パスワード、ポート(初期値4455)を確認
ライブモードのクリップの冒頭が欠ける	OBSの最大リプレイ時間を60秒以上に変更
F8、F9が動作しない	同じキーを使用する他のプログラムを終了して再起動

その他の問題はGitHubのIssueに登録。

12. 著作権とガイドライン



- クリップは元配信の二次創作物に該当し、
  原著作者と所属事務所のガイドラインの適用対象
- 書き出し設定のガイドライン一覧で事務所を選択すると、
  元配信のタイトルとURLを含む概要欄を作成し、確認項目を点検
- 確認項目: メンバー限定配信の該当有無、アーカイブの公開有無、
  チャンネル登録の有無、AI使用の表記
- 確認結果は参考資料であり、法的判断ではない。
  アップロード前に各事務所と配信者のガイドライン原文を確認

Stream in. Clips out.

stream2clip 取扱説明書。github.com/BudongJW/stream2clip