

DaVinci Resolve caption automator

2026-06-05

Transcribes a video locally and emits styled, position-correct captions as FCPXML that DaVinci Resolve imports natively.

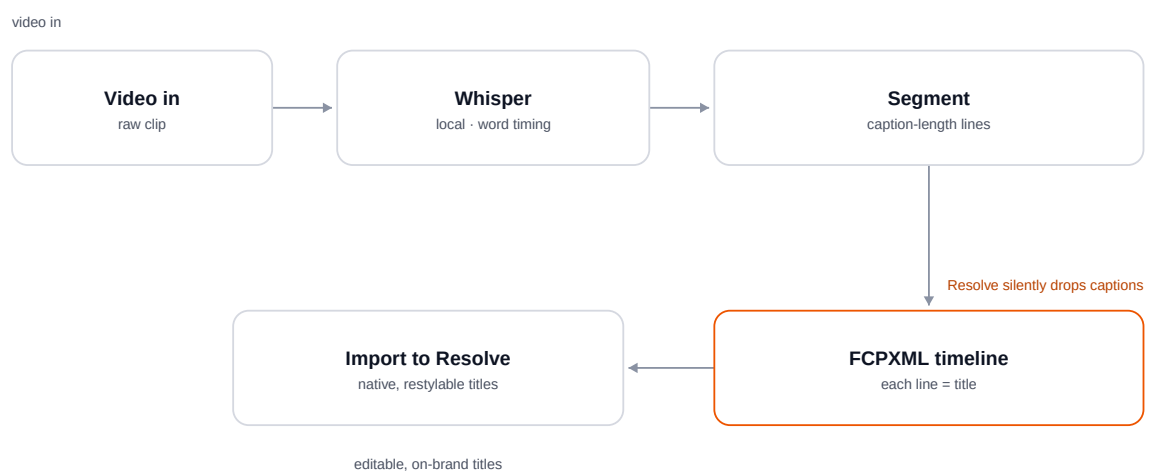
Python · Whisper · FCPXML · DaVinci Resolve

Problem

Captioning a long video by hand inside an editor is slow work, and the tools that promise to speed it up mostly export burned-in subtitles you can never restyle, or a format DaVinci Resolve chokes on. What an editor actually wants is captions that land in the timeline as real, editable title clips, already in the project's own style, ready to nudge and restyle like anything else.

Approach

The video is transcribed locally with Whisper for word-level timing, so nothing leaves the machine and there is no per-minute API bill. The transcript is segmented into caption-length lines tuned for how people actually read, then written out as an FCPXML timeline where every caption is a proper title clip pointing at the editing template. The one constraint that makes or breaks it: Resolve silently drops caption elements on import, so each line has to be emitted as a title instead. Once that is right, the whole timeline drops in cleanly.



Result

A raw video becomes an import-ready, fully captioned timeline in minutes, and the captions arrive as native Resolve titles you can restyle in one click instead of burned-in text you would have to redo. It runs across my client editing work and turns what used to be an evening of manual subtitling into a single import.

What I'd do differently

Add speaker diarization with per-speaker styling as an option for interviews and multi-person edits, and wrap the whole thing in a small interface so a non-technical editor can drop in a video and get the timeline back without ever touching the command line.

Repository: <https://github.com/hermann-ndamen/resolve-captions-fcpxml>

Interactive version and demo: <https://www.hermannndamen.com/portfolio/ai-engineer/resolve-captions-fcpxml>