DOCUMENTATION

User Guide

RELEASE DATA	
VERSION	2.0
RELEASED	June 2026
PLATFORM	Windows 10 (build 19041 "2004") or later, 64-bit

SUPPORT THE PROJECT

Support: free to use, no ads, nothing locked. If this app helps you, you can support its development at https://portablevideocompressor.ca/donate (optional).

01 What This App Does

Portable Video Compressor makes video files smaller while trying to keep quality as high as possible. You give it a size in MB; it does the re-encoding under the hood, adjusting bitrate - and, in Auto mode, resolution, frame rate, and the audio budget - until each file lands as close to your chosen size as it can, inside a leniency band you control. It works on one file or a whole queue per run. This guide walks through every control and page. Setup, requirements, and troubleshooting live in the README; how it all works internally lives in the Technical Specification.

02 Starting Out

Start the app - from the Start menu if you used the installer, or by running the `PortableVideoCompressor.exe` at the root of the portable folder (install notes are in the README). The main window shows two columns of cards - Files, Video Encoder, Format, Target Size, Resolution & Frame Rate, and the Start card - with the progress bar, run buttons, elapsed time, and status line along the bottom. Controls that don't apply right now are grayed out: everything configuring a run locks while one is active, and the manual fields lock while Auto mode is on. The title in the corner opens the project website; the Reset to Defaults button beside it is covered in section 17.

03 Adding Videos

- Click Select Input Video(s) for the standard file picker (multi-select works), or drag files from Explorer straight onto the window - the drop caption reads "Replace queue", and that's exactly what it does: dropping replaces the current selection, it doesn't add to it. To add to an existing queue instead, drop onto the Queue page (section 15).

- Dropped items are filtered to video files; folders and other file types are ignored. Accepted extensions: `.mp4` .mov .mkv .avi .wmv .flv .webm .m4v .mpg .mpeg .m2ts .ts .3gp .ogv .mts .vob .divx. A file already in the queue isn't added twice.
- The Files card shows your selection: a single path, or "N files selected (first: <path>)". During a run it switches to per-file form, "File 3 of 8: <path>" - with a * after the number when that file carries its own overrides (section 15.2). Long paths truncate in the middle; hover for the full path.
- The View Queue button lights up once you have two or more files.

04 The Output Folder

- By default, outputs go next to their source: the folder label follows your most recent selection. Click Change Output Folder to pick one folder for the whole batch; a folder you pick explicitly sticks, across new selections and across restarts, until Reset to Defaults returns it to the follow-the-input behavior.
- Finished files are named <name>_compressed with the container's extension, so nothing ever silently replaces your original. What happens when that name is taken is handled by a single prompt at start time (section 13.3).

05 Target Size and the Leniency Slider

- Target Size (MB) is the size each file aims for; it applies per file, not to the batch total. Blank means 10 MB. Individual files can override it (section 15.2).
- The Leniency slider (50-95%, default 80%) sets how much undershoot still counts as a hit. At 80% with a 10 MB target, the accepted band is 8-10 MB; the app aims for the middle of the band and stops as soon as an attempt lands inside it. A tight band (95%) means more re-encode attempts for a more exact size; a loose band finishes faster.
- Files that finish outside the band - under the floor, or over the target when nothing smaller could be produced - still keep their best attempt; the Queue page marks them amber instead of green (section 15.4).

06 Auto Mode

Auto mode (the default) makes the technical decisions per file:

- It splits the size budget between audio and video, keeping audio at or below its source rate, never below 64 kbps, and reserving the bulk for video.
- It then checks whether the video budget is enough for the source resolution and frame rate. If not, it steps the resolution down through standard tiers (2160, 1440, 1080, 720, 480) before it ever touches frame rate, and only then steps the rate down through common values (120, 90, 60, 50, 48, 30, 25, 24) - HEVC encoders are allowed to keep more than AVC at the same budget. It never upscales and never raises frame rate; a file that already fits keeps its source resolution and timing untouched.
- Frame rates below 24 are left alone rather than "corrected" upward.

- While Auto is on, the Height / Width / FPS fields and the Sample encode switch are disabled: Auto owns those choices (it turns sampling on by itself for clips of 60 seconds or longer), and 2-pass is forced on for the software encoders. You can still pin any of resolution, frame rate, or FPS for an individual file from the Queue page - Auto respects per-file overrides and only decides the axes you left alone.
- If even minimum audio alone can't fit the target for a clip's duration, that file fails up front with an "Impossible target" message instead of wasting an encode.

07 Manual Mode

Turn Auto mode off to take over:

- Height / Width: set both for a fixed resolution, set one and the other follows the source aspect ratio, or leave both empty to keep the source resolution. Odd values are snapped down to the nearest even number when you press Start - the fields update so you see exactly what will be used, and the status line notes the adjustment.
- FPS: a positive value re-times the video to that rate; empty keeps the source timing (no frame-rate conversion at all).
- Audio: your codec choice from the Format card applies as-is (section 10), and the same impossible-target check runs with the selected codec's real rate.
- Manual mode is also where the Sample encode and 2-pass switches are yours to set (section 12).

08 Choosing a Video Encoder

- The dropdown lists only encoders that actually work on this machine, verified by a quick test encode at every launch: "Software (AVC/H.264)" and "Software (HEVC/H.265)" run everywhere FFmpeg does; "GPU - NVENC (AVC/H.264)" / "(HEVC/H.265)" need an NVIDIA GPU; "GPU - AMF" variants need an AMD GPU. When a GPU encoder is available the app selects it by default, preferring NVENC H.264.
- Software encoders give the best quality per megabyte and support 2-pass; GPU encoders are several times faster and are the practical choice for big batches. HEVC (H.265) fits noticeably more quality into small targets than AVC (H.264), at the cost of playback compatibility on older devices.
- On older GPUs, the launch check may find that an encoder works only without its newest quality flags; the app then quietly uses that reduced profile as the machine's default for the encoder, so it works instead of erroring. The tuning page always shows the profile actually in use.

09 The Encoder Tuning Page

Click Encoder Tuning to edit the active encoder's profile. Each of the six encoders keeps its own profile, and edits apply from the next run - there is no separate save step.

- The pickers cover what the selected encoder supports: Preset for all of them (ultrafast through placebo for the software encoders; "P1: Fastest" through "P7: Slowest" for NVENC; quality / balanced / speed for AMF),

Tune where it exists, and Rate control and Multi-pass for the GPU encoders. Pickers that don't apply to this encoder say so and stay disabled.

- The options text box below is the profile itself - the exact FFmpeg flags used. Picker changes rewrite their one flag in the text; typing in the text moves the pickers to match. A value typed outside a picker's known list flips that picker to "Custom (edit text below)", and a note lists which pickers are custom. Everything commits as you type; Done and the Back arrow both just leave.
- Reset (on this page) restores this machine's detected default profile for the current encoder - including the reduced profile on older GPUs (section 8) - not a generic factory string that might not run here.
- If a run fails and your profile differs from the machine default, the failure message says so and points you at Reset first.

10 Audio and Container

Two pickers on the Format card, filtered against each other so an invalid pair can't be selected:

AUDIO OPTION	WHAT IT DOES
Same as source (no re-encode)	copies the original audio track unchanged (default)
AAC	re-encodes; the modern, widely compatible choice
MP3	re-encodes; maximum legacy compatibility
WAV (PCM 16-bit)	uncompressed audio; large, and not allowed inside MP4

CONTAINER	NOTES
MP4	the default; starts playback fast when streamed
MOV	QuickTime; same fast-start behavior as MP4
MKV	carries practically any audio, including copies MP4/MOV refuse

- "Same as source" is checked per file against the real source track: some audio types (FLAC, Opus, PCM and others) can't be carried by MP4 or MOV. Such a file fails fast - before any encoding - with a message that says which audio it found and names your options: re-encode the audio, or switch to a container that accepts a copy (MKV always does).
- Re-encoded audio budgets: in Auto mode the app decides the rate (section 6); in manual mode AAC and MP3 use sensible defaults and WAV uses its full uncompressed rate, which the impossible-target check takes into account.

11 Preserve Source Color

- Off (the default), every encode converts to the most compatible output there is: 8-bit 4:2:0, standard BT.709 color. That plays everywhere and is the right choice for screen recordings, phone footage, and general use.

- On, the source's own color format passes through - wide gamut, 10-bit, HDR metadata, 4:2:2 chroma - for footage where that matters. Two costs: the same target size now has to carry more data per pixel (the pre-run quality check accounts for this), and consumer GPU encoders can't ingest professional 4:2:2 10-bit sources at all - those files need a software encoder, or the switch off.
- The switch applies to all encoder profiles at once, as its caption says. It reflects the profiles' actual state: hand-editing color flags on the tuning page is respected, and with mixed hand-edited profiles the switch reads Off.

12 Sample Encode and 2-Pass

Two ways to spend a little time for a more accurate size:

- Sample encode runs a short mid-clip test encode first (4-12 seconds, taken from the middle where the content is representative) and uses its measured size to calibrate the starting bitrate - usually saving one or two full re-encode attempts. It applies to clips of 10 seconds or longer; in manual mode the switch decides, and in Auto mode it turns itself on for clips of 60 seconds or more.
- 2-pass (software encoders only) analyzes the whole file first and then encodes with that knowledge - better quality distribution and a more predictable size, for roughly double the encode time. The progress bar shows the analysis pass as the first half. Your manual choice is remembered as a preference: when Auto mode or a GPU encoder forces the switch, your preference is untouched and comes back the moment a supporting encoder is selected again.

13 Starting a Batch

Press Start Compression here or on the Queue page. Before any encoding, in order:

13.1 Validation

Your target and any manual fields are checked (section 7 covers the even-value snapping). Bad values stop with a specific message; nothing has run yet.

13.2 Two quick sanity checks

- If every selected file is already at or under its target, the run ends immediately with "Nothing to do" - re-encoding those files could only make them bigger.
- The batch's worst-case output size, plus a safety margin, is compared with free space on the output drive; if it doesn't fit you get the numbers and a Continue / Cancel choice.

13.3 The overwrite prompt

If any planned output name is taken - a file already on disk, or two queued videos that resolve to the same name - one dialog lists the conflicts (up to 8, each annotated with why) and asks once for the whole batch: Yes overwrites; No auto-renames with a numeric suffix (`video_compressed1.mp4`, `video_compressed2.mp4`, ...) so

nothing existing is replaced.

13.4 The quality warning

Every file gets a quick estimate of the smallest size that can still look reasonable at its settings; files whose target is below that estimate are listed in a dialog before anything encodes. Each row shows the estimate ("Minimum passable: ~N.N MB") next to an editable Target box pre-filled with the file's current target:

- Untick a file to leave it out of this run (it's recorded as skipped, not forgotten).
- Raise a target right in the dialog; changed values are applied to those files as per-file overrides, immediately visible on the Queue page.
- Continue proceeds (invalid targets are pointed out and block it); Cancel aborts the whole run before any encoding.

Ignoring the warning is allowed - it's an estimate, and you may prefer a smaller, softer file.

---

14 During the Run

- The status line narrates the current file: "Analyzing input video...", "Running sample encode for bitrate estimation...", "Encoding video...", "Retrying with adjusted bitrate...". The progress bar covers the current file (for 2-pass, the analysis pass is the first half). Files already small enough are announced as skipped and the run moves on.
- Pause / Resume freezes the encoder process at the operating-system level - no work is lost, nothing re-encodes, and the elapsed clock stops counting while paused. Pausing between passes or between files works too: the next process starts frozen.
- Skip File abandons the current file immediately (no confirmation) and cleans up its partial output completely; the batch continues.
- Stop Batch asks for confirmation, tells you how many files would be left undone, and keeps everything already finished. A stopped run's files never end in half-written outputs - a cancelled file leaves nothing behind.
- Time elapsed excludes paused time. While a run is active (from the moment you press Start), the configuration controls and the queue lock; drops onto either window are refused until it ends.
- Closing the app mid-run stops the encoder cleanly; nothing keeps running in the background.

---

15 The Queue Page

View Queue opens the batch manager - a full list of your files, and the live dashboard while a run is active.

15.1 Rows

Each row shows its position and filename, the full path, a size line ("Input:

128.4 MB", growing to "Input: 128.4 MB -> Output: 9.6 MB" once done, plus

"Target: N MB"), and a processing-time line ("Elapsed:" ticking for the running file, "Took:" once finished). The header states the file count and how many carry overrides.

15.2 Per-file overrides

Tap a row to expand its editor. Every field starts at "(batch default)"; set only what should differ for this file:

- Target size (MB), Width, Height, FPS - numbers, same rules as the main window (Auto mode respects pinned axes and decides the rest, section 6).
- Video encoder, Audio codec, Container - pickers with the same choices as the main window; the audio/container pair filters itself the same way, and switching a file's container changes that file's output extension.
- "Reset this file to batch defaults" clears the row's overrides; Clear All Overrides in the footer clears every row (with a confirmation).

Overridden rows get a marker next to their name, and the main window's per-file label carries a * for them during runs. Overrides live with the queue: they aren't saved as defaults and they don't survive replacing the selection.

15.3 Managing the list

- Reorder by dragging a row's handle (the dotted grip); the list auto-scrolls when you drag near the top or bottom edge. Files process in list order.
- The X on a row removes that file (with a confirmation). Dropping videos from Explorer onto this page appends them to the queue - the caption reads "Add to queue" - unlike the main window, which replaces.
- Editing, reordering, removing, and dropping all lock while a run is active; the page switches to its live read-only view.

15.4 The live view and result tints

During and after a run the rows tell the story at a glance:

- The running file is highlighted in the accent color, with its own progress bar and status right in the row.
- Green: finished inside the leniency band. Amber: finished, but outside the band - smaller than your leniency floor, or over target when nothing smaller could be made. Red with a strikethrough: failed (the reason is in the summary and the error log). Gray with a strikethrough: skipped or cancelled.
- Tints and per-row targets are frozen with the values in force when each file finished - moving the slider or changing targets afterwards never recolors a finished row. Results stay visible until you start another run or replace the selection.

The footer mirrors the main window's run controls, including Start, so a batch can be managed entirely from this page.

16 After the Run

One summary dialog counts processed, succeeded, failed, and skipped files (your quality-dialog skips and the already-small ones are counted separately), shows the space saved, lists up to five failure reasons inline, and offers to open the output folder. When anything failed, the full details - including FFmpeg's own error output - are in `PVC_batch_errors_<timestamp>.txt` next to the outputs.

17 What the App Remembers

Saved automatically as you work, restored on the next start: target size, Auto mode, the manual Height/Width/FPS entries, audio codec and container, chosen encoder, the Sample switch, your 2-pass preference, Preserve source color, leniency, an explicitly chosen output folder, every encoder's tuning profile, and the window position. Per-file overrides are deliberately not saved - they belong to a batch, not to your defaults. Reset to Defaults (top right, confirmation-gated, unavailable mid-run) clears all of that in one step: fields blanked, Auto and Sample and Preserve color off, leniency back to 80%, pickers back to "Same as source" + MP4, every tuning profile restored to this machine's detected default, the output folder back to following the input, and hardware detection re-run. Per-file overrides on a loaded queue are left alone.

18 Going Deeper

The README covers setup, upgrading the bundled FFmpeg, and troubleshooting. The Technical Specification documents the internals - the exact math, commands, and file-by-file structure - if you want to know precisely what happens under the hood.

END OF USER GUIDE