



DOCUMENTATION

README

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 Is

Portable Video Compressor (PVC) shrinks video files to a size you pick, in MB. You choose the target; the app works out the bitrate, encodes, measures the result, and re-encodes at an adjusted bitrate until the file lands inside your accepted range. It handles one file or a whole queue per run. The real encoding work is done by two bundled command-line tools:

- [ffmpeg.exe](#) does the video and audio encoding.
- [ffprobe.exe](#) reads media facts (duration, resolution, frame rate, audio).

Everything else - the math, the retries, the queue, the safety checks - is the app. This file covers setup and problem-solving; the User's Guide covers every control and workflow; the Technical Specification explains the internals.

02 Requirements

- Windows 10 64-bit, build 19041 (version 2004) or later - including Windows 11.
- A 64-bit CPU and enough free disk space on the output drive for the finished files plus temporary attempt files made during encoding. The app checks this before each run and warns you if the worst case would not fit.
- Optional: an NVIDIA GPU (NVENC) or AMD GPU (AMF) for much faster hardware encoding, provided the bundled FFmpeg build includes support for it. No GPU is required; the software encoders run everywhere.

03 Installing

PVC comes in two downloads; the app inside is the same either way. Portable (the downloaded folder):

- Unzip the folder anywhere writable (C:\Tools\PortableVideoCompressor, a USB drive, wherever) and run the [PortableVideoCompressor.exe](#) at its root. There is no setup step, and nothing is written outside the folder - no registry entries, no Start menu shortcut (make your own if you want one).
- The root stays tidy: the launcher exe, [portable.dat](#), [settings.json](#) (created on first save), and the [FFMPEG](#), Documents, and Licenses folders. The application itself lives in App - the root exe is a small stub that starts [App\PortableVideoCompressor.exe](#), so the folder can be moved or renamed freely.
- The [portable.dat](#) marker at the root is what makes this copy portable: [settings.json](#) is stored beside it, so moving the folder moves your configuration with it.
- [FFMPEG](#) sits at the root so upgrading it stays a simple file swap (section 9).

Installed (the setup program):

- Run the setup executable and follow it. You get a Start menu entry, and the app shows up in Windows' installed-apps list with a normal uninstaller.
- The installed copy has no [portable.dat](#) marker, so it stores [settings.json](#) under your Windows profile at %AppData%\PortableVideoCompressor - the install location itself is usually not writable.

Either way the settings file is ordinary JSON you can read, back up, or delete to start fresh.

04 First Run

- On every launch the app test-encodes one second of synthetic video with each hardware encoder to find out what actually works on this machine (software AVC is always available). Only working encoders appear in the Video Encoder dropdown. No window flashes and no files are written; a healthy check takes a few seconds.
 - Each hardware encoder is tried at up to three capability levels: the full quality profile first, then a version with the newest GPU-only flags removed, then a bare minimum. The best level that works becomes that encoder's default here - so an older GPU still gets a working profile instead of failing on flags it doesn't support. Details are in the Technical Specification.
 - If a second copy of the app is launched, it brings the already-running window to the front instead of opening twice.
 - The window follows your Windows light/dark theme and accent color, and switches live when Windows does. It opens at its fitted size and remembers only its position between sessions - never a stale size.
 - The app title in the corner is a link; clicking it opens portablevideocompressor.ca in your browser.
-

05 Basic Usage

- Click Select Input Video(s), or drag video files onto the window. Multiple files become a queue processed in order.
- Confirm the output folder. By default it follows the folder of your most recent selection; once you pick a folder explicitly, that choice sticks (including across restarts) until you use Reset to Defaults.

- Enter the Target Size (MB), e.g. 10.
- Leave Auto mode on for automatic resolution, frame-rate, and audio budgeting, or turn it off to set those yourself.
- Pick a Video Encoder. Software (AVC/H.264) is the safest, most compatible choice; the GPU options are much faster when your graphics card supports them.
- Set the Leniency slider to the undershoot you will accept: at 80% with a 10 MB target, anything from 8 to 10 MB counts as a hit.
- Click Start Compression. Watch the per-file status and progress at the bottom, or open the Queue page for a live per-file view.

Outputs are named <name>_compressed.<ext> next to your videos (or in your chosen folder). If an output name is already taken - on disk, or by two queued files wanting the same name - one prompt before the run asks whether to overwrite or auto-rename with a numeric suffix ([video_compressed1.mp4](#)). The full walkthrough of every card, switch, and page is in the User's Guide.

06 What Runs Before Encoding

Start is not a blind fire. In order, the app:

- validates your target and any manual fields (odd widths/heights are snapped down to even, and the fields update so you see what will be used);
 - skips the run entirely if every selected file is already at or under its target ("Nothing to do");
 - estimates the finished size of the batch and warns you when the worst case, plus a safety margin, exceeds the free space on the output drive;
 - resolves output-name collisions with the overwrite/rename prompt;
 - estimates, per file, the smallest size that can still look reasonable at your settings, and lists any file whose target is below that estimate in a quality warning dialog - where you can untick files to skip them or raise their targets on the spot before continuing.
-

07 Controls and Status

- Start Compression begins the run. Pause / Resume freezes and resumes the FFmpeg process at the operating-system level, so no work is lost and nothing re-encodes; the elapsed clock stops while paused. Skip File abandons the current file (its partial work is cleaned up) and moves on. Stop Batch asks for confirmation, tells you how many files remain, and keeps everything already finished.
 - The progress bar, elapsed time, and a one-line status track the current file ("Analyzing input video...", "Encoding video...", "Retrying with adjusted bitrate...", and so on).
 - After the run, a summary counts successes, failures, and skips, shows the space saved, and offers to open the output folder. When failures exist, [PVC_batch_errors_<timestamp>.txt](#) is written next to the outputs with each file's reason - including FFmpeg's own error lines, which never clutter the dialog.
-

08 Troubleshooting

- "FFmpeg not found": the app could not locate `FFMPEG\ffmpeg.exe` and/or `FFMPEG\ffprobe.exe`. Restore the `FFMPEG` folder at the root of the application folder (beside App) and try again.
- "Impossible target": the target size cannot hold even minimum-bitrate audio for the clip's duration. Increase the target or trim the source.
- Output file locked: if the existing output is open in another program when the app tries to replace it, that file is reported Failed - and the best encoded attempt is kept next to it under its temporary `_tmpN` name, so the work is not lost. Close the other program and rename the kept file.
- GPU encoder missing or reduced: update your GPU drivers, and confirm the FFmpeg build includes NVENC/AMF. On older GPUs the app deliberately selects a conservative default profile for that encoder; Reset all settings (or the tuning page's Reset) re-detects on demand.
- A file fails only with "Preserve source color" on: professional 4:2:2 10-bit footage is a format consumer GPU encoders cannot ingest. Use a software encoder for those files, or turn Preserve source color off.
- Unreadable / corrupt input: media probes are time-limited; a file that cannot be read fails cleanly with a per-file message instead of hanging the run.
- Files come out under your leniency range, or over target: the Queue page tints those rows amber instead of green; the run still kept the closest attempt. Raising leniency widens what counts as a hit.

09 Upgrading the Bundled FFmpeg

The app treats FFmpeg as a drop-in dependency. To upgrade:

- Close the app and replace `FFMPEG\ffmpeg.exe` and `FFMPEG\ffprobe.exe` with the new 64-bit Windows builds.
- Start the app - encoder detection re-runs automatically at every launch, so the encoder list and per-machine default profiles re-derive against the new build. "Reset all settings" also re-detects on demand.
- Run a short test batch before relying on the new build.

Compatibility note: the default NVENC profiles use `-rc vbr_hq`, which upstream FFmpeg has deprecated in favor of `-rc vbr` combined with the newer preset/tune system. The bundled build accepts it; if a future FFmpeg removes the flag, the automatic tier probe falls back to a working profile on its own, and the tuning page lets you adopt the modern flags explicitly at any time.

10 Uninstalling, License and Credits

To remove the portable version, delete its folder - nothing else exists anywhere on the system. To remove the installed version, uninstall it from Windows' installed-apps list like any other program; your settings folder at `%AppData%\PortableVideoCompressor` stays behind unless you delete it too. Portable Video Compressor is built on .NET MAUI / WinUI and on FFmpeg (`ffmpeg.exe` and `ffprobe.exe`), which are licensed separately under



their respective licenses; see the included Licenses folder and the FFmpeg project website for details.

 END OF README