PLUGIN MANUAL · FOR 0.4

PowerDuck - Quick Start

Automatic audio ducking for **Final Cut Pro** on the **Mac** (Audio Unit + VST3, also runs in Logic and any AU/VST3 host). One copy listens to your dialogue, one rides the gain on your music or SFX, staged, ducked and EQ-pocketed to broadcast practice, automatically, with a slider for everything. macOS only.

Watch Tutorial

The whole setup in one sitting: the pair, Auto Detect, the settings that matter, and the duck happening live on a real project.



Install & quick start

Run the installer: it places both plugin formats into the system folders every host scans:

- **Audio Unit** → /Library/Audio/Plug-Ins/Components/PowerDuck.component
- **VST3** → /Library/Audio/Plug-Ins/VST3/PowerDuck.vst3

Then **restart Final Cut Pro**: it scans for new plugins at launch.

1 **Voice on the dialogue**
Effects browser (**⌘5**) → search **PowerDuck** → drag it onto your dialogue clip. It starts in **Voice** mode. Done; it listens and passes audio through untouched.

2 **Music / SFX on the music**
Drag PowerDuck onto your music clip, open its panel (controls icon in the Audio inspector) and set **Put this on → Music / SFX**.

3 **Click a Quick Style, play**
Pick **Podcast, YouTube, Documentary, Music-Forward** or **Cinematic**. Press play: the chips read **"Linked: hearing Voice"** and the music ducks.

Many clips? Best results: run PowerDuck as part of your final mix, on a compound clip. One Voice on a compound clip hears the whole conversation as a single continuous clip, and one pair covers everything inside it. In a hurry, set up one Music/SFX copy the way you like, then copy the clip and **Edit → Paste Effects** (⌥⌘V) onto the rest.

The pair & Link Groups

Final Cut Pro has no mixer buses and no sidechain routing, so PowerDuck works as a **linked pair**: instances find each other automatically through a shared, timeline-synced link.

- **Voice**: put on dialogue. Publishes what it hears; audio passes through untouched. The default mode, so a stray drop is always harmless.
- **Music / SFX (Duck)**: put on anything that should get out of the voice's way. Reads the voice level at its own timeline position and rides its gain.
- **Auto (Detect)**, the third pill: click it and PowerDuck listens, classifies the clip as Voice or Music, sets the pill for you and **locks the answer**: it never re-guesses afterwards. With Final Cut's *Background Render* on (Preferences → Playback) the verdict can land the moment you drop the plugin; with it off, on the first play. If eight seconds of clearly active audio pass without a confident read it settles on Voice, which is always harmless. Picking Voice or Music yourself overrides it any time; **Reset** re-runs detection.
- **Link Group (1–4)**: copies with the same group number talk to each other. Several ducked instances can share *one* group with different amounts (music −12, ambience −6); separate groups are only needed for *different voices* (say, narrator vs. interview).

Many clips at once

Dialogue cut into dozens of clips? Set up PowerDuck on **one** dialogue clip, copy it (⌘C), select all the other dialogue clips, then *Edit* → *Paste Effects* (⌥⌘V): Final Cut applies the effect with all its settings to every selected clip in one go. (Use *Paste Attributes…* ⇧⌘V to pick just PowerDuck if the source clip

carries other effects.) Do the same once for your music and SFX clips with a configured Music / SFX copy. A whole timeline takes under a minute. Just remember clips added *later* need a paste too, which is exactly what compound clips (below) solve.

> **But for anything beyond a quick job, compound your dialogue instead (next section).** Every pasted copy is its own plugin instance that sees only its own clip. The timeline features (Duck Ahead, Rise Ahead, Bridge Gaps) read the learned timeline, and they are at their best when *one* Voice hears the whole conversation as one continuous piece. That is exactly what a compound clip gives them.

Best practice: the final mix, on a compound clip

PowerDuck is best used as part of the final mixing and audio work, applied to a compound clip. Cut the story first; when the edit has settled, compound it and mix. On a timeline of loose individual clips, every clip carries its own plugin copy, each copy sees only its own slice of the story, and every re-arrange moves the ground under the learned timeline. Inside a compound at mix time, *one* Voice hears the entire dialogue as one continuous, stable clip: the role is detected once, the learned timeline has no seams, and Duck Ahead and Rise Ahead can see across every cut. It is as close to a real mix bus as Final Cut gets, and if ducking ever feels inconsistent on individual clips, compounding them at mix time is the single biggest reliability upgrade you can make.

1. Make sure the clips inside your compound carry the right roles (**Dialogue** / **Music**, usually automatic).
2. Select the compound clip → **Clip → Expand Audio Components** (^⌥S). It unfolds into one lane per role.
3. Apply PowerDuck to the **Dialogue** component → Mode **Voice**.
4. Apply PowerDuck to the **Music** component → Mode **Music / SFX**. Collapse again, done.

> **Why this is the best way:** it's two instances total, and anything you add inside the compound later automatically joins the duck as long as its role is right. If FCP shows components split by *subrole* ("Dialogue-1", "Dialogue-2"), either switch the grouping to parent roles or put a Voice copy on each dialogue component: multiple Voices on one group combine safely.

Auto: Match Voice

The two **Auto (Match Voice)** switches turn PowerDuck from a ducker into a mixer. Both measure how loud your voice actually is and place the music against it, using standard broadcast balances. **Both ship on Auto**, so out of the box PowerDuck ducks intelligently and places the bed for you. Switch Bed Level to Manual when your own clip volumes should rule and the only thing PowerDuck does is duck:

- **Bed Level: Auto**: positions the *unducked* music against your voice: it rides the music down from Music Level to sit ~10 dB under the speaker (the "no-one-is-speaking" balance). Music Level stays the hard top.
- **Duck Depth: Auto**: during speech, ducks just enough to hold the music at your chosen **Voice Priority** under the voice: **Gentle** 16 dB, **Standard** 20 dB (documentary), **Strong** 24 dB (podcast/YouTube: every word lands). Loud music gets pushed harder; an already-quiet bed is left alone.

Both Autos work like a mixer's hand, not a meter: the depth and bed are *set for the phrase and held*, re-aiming only between phrases, so music with a heavy beat cannot pump the level while someone is talking.

Prefer a fixed amount? Set Duck Depth to **Manual** and use the **Duck Amount** slider (8–18 dB covers most real mixes).

> **Tip:** Auto reacts to the dialogue as it is. If your dialogue levels are wildly uneven, level them first (that's PowerEQ's job), then Auto ducking locks in beautifully.

Quick styles

One click sets depth mode, priority, duck style, bed level, cut width and all three fade times.

Podcast
Voice is king: full auto, strong priority, tight broadcast moves.

YouTube
Voiceover-first: full auto, strong priority, fast moves, wide clarity cut.

Documentary
Balanced storytelling: full auto, standard priority, smooth moves.

Music-Forward
Keeps the bed alive: full auto, gentle priority, long easy fades.

Cinematic
Trailer feel: fixed 9 dB full-band duck, slow dramatic moves.

The duck history strip

Reading it: dim bars are the music as staged, bright bars are what actually plays, the **orange line** is the volume ride being applied (dipping under speech, recovering after), exactly the curve Bake exports as keyframes, and the orange shading marks the slice the duck removes. Yellow ticks along the top are voice activity.

The strip in the panel shows the **last 8 seconds as they played**: the music's level (dim bars) with the actually-applied duck carved in (bright bars), and yellow ticks along the top wherever the detector heard voice. The readouts show the current gain, depth, bed correction and the ceiling limiter.

Why doesn't the clip waveform in the timeline show the duck? Final Cut draws clip waveforms from its own internal snapshots. No plugin can make the timeline waveform track a live cross-clip link (that's true of every ducking plugin). The history strip is the truthful readout: it records exactly what was applied. Your ears, the strip and the export always agree.

The yellow ticks are also the fastest way to tune the **Voice Threshold**: if ticks appear during room tone, raise it until they only light up on speech.

Why the timeline waveform doesn't change

A very common first impression: you press play, the ducking clearly *works*, but the music clip's waveform on the timeline looks exactly the same, so it feels like nothing happened. **The audio is ducked; the picture is stale.**

Final Cut draws clip waveforms from its render cache and only redraws them when *it* decides something changed: a parameter edit, a clip move. PowerDuck's ducking comes from the linked Voice through a side channel Final Cut's cache knows nothing about, so the waveform you see is usually from a render pass made *before* the pair ever linked. This is a Final Cut drawing behavior, not a PowerDuck level problem.

- **Trust your ears and the Duck History strip**: the strip in the PowerDuck panel is the live, accurate picture of exactly what the duck is doing.

- **Want it visible on the timeline?** Use Bake Keyframes: the ducking becomes ordinary volume keyframes on the clip, which Final Cut draws (and you own) like any hand-made mix.
- Sometimes Final Cut *does* re-render a clip through the effect after you've played a section, and the waveform suddenly shows the dips: that's the same caching behavior working in your favor. Either way, what you hear is the truth.

Controls

Top of the panel

- **Put this on**: **Auto (Detect)**, **Voice** (dialogue) or **Music / SFX** (anything that ducks). Auto classifies the clip once, sets the pill and locks. See the pair.
- **Link Group**: which Voice this copy listens to (1–4).
- **Before / After**: press and hold to hear the mix without ducking.
- **Status chips**: "Linked: hearing Voice", "Waiting for Voice…", or "Idle: press play", plus the group.

Ducking

- **Duck Depth**: Auto (Match Voice, the default) with **Voice Priority** Gentle / Standard / Strong, or Manual (fixed **Duck Amount**).
- **Duck Style**: **Smart (Clarity)** splits the duck: a gentler overall dip plus a dynamic EQ cut at ~2.8 kHz where speech intelligibility lives, so the voice floats while the music keeps its body. **Full-Band** is the classic radio duck.
- **Cut Width**: how wide the Smart clarity cut is (narrow = surgical, wide = maximum voice space; Smart style only).

Music level

- **Bed Level**: **Auto (Match Voice), the default**: PowerDuck places the bed against your voice, with Max Level as the hard top. Switch to **Manual** to keep PowerDuck out of the level path entirely: your clip volumes are the mix and the only thing the plugin does is duck.
- **Music Volume**: *Bed Level Manual only.* How much to bring the music **up where there is no voice**: the gaps between lines, and the stretches with no dialogue at all. It rides in on your Fade Up and is fully out underneath speech, so it only ever fills the holes and never touches the duck. Default 0 dB, which keeps Manual a bit-exact null.
- **Max Level**: *Bed Level Auto only.* The music's level, one slider with two clear roles. **With no voice around** (gap clips, intros, music-forward sections, a solo music clip) it is the ground truth: quiet tracks are lifted to it, hot tracks held at it with a slow, transparent hold (not a pumping limiter) and the

music plays at exactly this level. **While a voice is present** with Bed Level on Auto, the bed sits the right distance under the voice automatically and this slider is simply the hard top nothing exceeds. Default −15 dB, safely under typical dialogue.

Timing (advanced)

The rule for the whole section: **the Ahead sliders say *when* a fade starts; the Fade sliders say *how long* it takes.** The two pairs mirror each other: Fade Down with Duck Ahead going into the voice, Fade Up with Rise Ahead coming out of it.

- **Voice Threshold**: how loud the dialogue must be to count as speech. Raise it if room tone triggers ducking; lower it for quiet speakers.
- **Fade Down** (20 ms – 2 s, default 250 ms): how long the duck's descent takes, at one steady rate.
- **Duck Ahead** (0 – 3 s, default 200 ms): how far *before the first word* the Fade Down starts, read from the learned timeline (a never-seen section ducks at the word instead). Set it at or above Fade Down and the music is fully down, resting, before the voice enters. Set it shorter and the fade is still mid-descent as the line begins: it simply keeps fading *under* the speech until it lands. Example: Fade Down 2 s with Duck Ahead 700 ms starts the glide 0.7 s ahead and completes it 1.3 s into the line.
- **Fade Up** (100 ms – 3 s, default 300 ms): how long the recovery takes, at one steady rate.
- **Rise Ahead** (0 – 2 s, default 0): how far *before the end of a phrase* the Fade Up starts, so the music can bloom under the last words the way a mixer rides a fader up early. Example: Rise Ahead 400 ms with Fade Up 2 s starts the swell under the final word and reaches full level 1.6 s after it. Endings come from the learned timeline and are verified genuinely silent for the whole window before a rise is allowed. A short line, or a pause with more speech coming, keeps classic timing (fully ducked to the end, rise after), so nothing is ever swallowed.
- **Bridge Gaps** (0 – 8 s, default 3 s), the look-ahead: pauses with more speech coming within this window hold the duck *rock-steady*: no volume or EQ motion at all until the next line lands. When there is *no* voice within it (a gap clip, a music-forward section, the end of an interview), the music rises at your Fade Up right after the last word. On sections PowerDuck has not learned yet the rise is a gentler, accelerating glide instead (so an unknown pause never pumps); any earlier play counts as learning, and a bake always sees everything. 0 = off.
- **Hold** (0 – 2 s, default 0): extra time the duck lingers after speech before any recovery begins. Bridge Gaps carries pauses from the learned timeline now, so Hold is purely a matter of taste.

Blue dots A/B individual sliders without losing your setting; double-click a slider's name to reset it. Hover any control's name for plain-English help.

Ignore (Music Rides): let the music win

On the **Voice** side there is a per-clip switch: **This Clip: Duck Under It** (the default) or **Ignore (Music Rides)**. Set to Ignore, the clip's dialogue still plays exactly as before, but the music does *not* duck for it: it rises to Music Level as if the clip were not speaking. Use it for montage narration you want washed in music, walla, or any moment where the music should carry.

Stacked clips resolve in the ducking's favor: if a Duck Under clip and an Ignore clip overlap in time (a connected interview bite over an ignored crowd bed, say), the music always ducks. An explicit "duck under me" beats an "ignore me", every time.

Ducking SFX & ambience

Despite the name, the Music / SFX mode ducks *anything*: sound effects, ambience beds, nat sound. Put a copy on each, keep them on the **same Link Group** as your Voice, and give each its own amount: a classic documentary setup is music −12 dB and ambience −6 dB under narration. Separate groups are only needed when different background beds should follow *different voices*.

Exports & stems

- **Share / export:** the duck renders exactly as you heard it: PowerDuck is engineered around Final Cut's render pipeline (cuts, prefetch, background passes) so playback and export always match.
- **Role-stem exports:** exporting the *Music role alone* produces the music **without ducking**: with no dialogue in that render pass there's nothing to duck against. That's usually exactly right for M&E/dubbing stems; for a ducked music stem, export the full mix with the dialogue role muted *in the mix* rather than as a solo stem.
- **Latency:** constant and tiny (~1.5 ms), reported to the host: nothing drifts, in either mode.

Bake to volume keyframes

Prefer the ducking as plain **volume keyframes on the clip**: no plugin needed at playback, visible in the timeline waveform, editable by hand? Press **Bake Keyframes** (bottom of the Music / SFX panel). One click does the whole job:

- If PowerDuck already has the dialogue captured, the clip's full volume curve is computed *instantly*: the same duck law the plugin renders with, so the keyframes match what you've been hearing.
- The first bake runs a short background analysis: the button shows **"Analyzing… span · %"** while PowerDuck asks Final Cut's render engine to sweep the linked clips (no playback, no export) and bakes automatically when the numbers settle. Click again to bake immediately with what's captured.

- **Watch the span.** PowerDuck can only bake the part of the clip Final Cut has rendered, and Final Cut renders where the skimmer or playhead *lingers*. If the span reads 0:49 on a 12-minute clip, hover a moment over the clip's very start and very end (⇧S audio skimming on) and Bake again; the full span bakes instantly. In a normal edit session you'll have played the timeline anyway: coverage accumulates all session, so by mix time Bake is usually already complete.

Then two steps in Final Cut Pro:

1. Drag the ⇢ **Drag into Final Cut** chip (bottom of the panel after a bake) straight into FCP's browser, or use **Open in Final Cut Pro** in the bake dialog and confirm the import. The baked "donor" clip is also saved to Desktop/PowerDuck Keyframes.
2. Click the imported donor clip, copy it (⌘C), select your music clip → *Paste Attributes…* (⇧⌘V) → tick **Volume** → Paste.

That's it: PowerDuck **bypasses itself** on the clip when it bakes (the pasted keyframes *are* the duck; Reset re-enables the plugin if you change your mind).

The keyframes are aligned to this clip's start, so paste lands them exactly where the duck happened. Volume keyframes carry level moves only: on the **Smart (Clarity)** style the dynamic EQ pocket can't ride along, so the bake carries the full depth as plain volume instead. Re-arranged the edit since baking? Just Bake again, and *Presets* → *Clear Bake Capture* starts the capture fresh after a big re-shuffle. Why can't the last paste step be automatic? Final Cut's clip clipboard is a private Apple format and plugins have no API to edit your clips: the donor + Paste Attributes route is the supported way in. If the background analysis can't wake Final Cut's renderer on your system, hovering the skimmer over the clips' starts and ends (⇧S audio skimming on) teaches PowerDuck the positions, and a play-through or one quick export still guarantees 100% voice coverage: the bake dialog always reports the exact percentage.

How the pros use it

Professional dialogue mixers don't ride every fader by hand and they don't trust a dumb gate either. The working method is: *automate a first pass, then own the exceptions*. PowerDuck is built around exactly that loop:

1. **Let Auto do the pass.** Pick a Quick Style, play the timeline once. Auto depth and bed leveling stage the music against the actual voice, Duck Ahead starts each move before the first word the way a mixer pre-rolls the fader, and Bridge Gaps keeps pauses down when more speech is coming.
2. **Read the strip, not just your ears.** The orange ride line is the exact volume automation being applied. Yellow ticks confirm what triggered it; "at max", bed and LIMIT readouts tell you what the staging is doing.
3. **Scene-by-scene overrides, natively.** Every PowerDuck control is a normal plugin parameter, so Final Cut can keyframe it in the inspector: drop Duck Amount for a montage, raise Voice Threshold for a noisy location scene. No extra instances needed.

4. **Finish like a mixer: Bake and hand-tweak.** When the edit is locked, Bake writes the whole pass as real volume keyframes on your music clip. Nudge any moment by hand in the timeline: the final ride is yours, visible in the waveform, no plugin required at playback.

That's the same division of labor a broadcast mix room uses: machines do the consistent 95%, a human decides the moments that matter.

Trial & license

Updates & ownership. Your license is permanent. Every update through version 1.x is free, including macOS compatibility fixes. A future major version will be a paid upgrade, always discounted for existing owners. If this product is ever retired I will say so rather than go quiet: the final version stays downloadable and your license keeps working.

- The **7-day free trial** starts the first time the plugin runs: full functionality, no account.
- Press **Setup** (top-right of the panel), paste your key, hit **Activate** (internet required once).
- **Plugin can't reach the internet?** Firewalls, VPNs and strict company networks sometimes block plugins while the browser works fine. When activation fails that way, the panel offers **Activate in your browser instead**: the browser page checks your key and hands you a signed code to paste back into the panel. After that the plugin never needs the internet again.
- When the trial ends, audio passes through unprocessed until a key is added: your projects keep playing, nothing is ever silenced.
- Keys come from payhip.com/b/p0nCa; a key covers about two machines.

"Lowering the security settings for Final Cut Pro"

Some builds between 0.5.1 and 0.7.0 made Final Cut show this when you added PowerDuck: *"Use of the requested Audio Unit(s) requires lowering the security settings for Final Cut Pro."* That was a packaging mistake on our side, and it is fixed: **0.7.1 and later load with no prompt at all**. Update and it is gone.

What it actually meant, since the explanation is hard to find anywhere: Final Cut runs audio plugins inside its own sandbox, a protective box that limits what a plugin can reach. A plugin has to declare that it can live in that box. PowerDuck always could, but one build forgot to say so, and rather than refuse to load it, Final Cut offered to run plugins outside the box instead. That is what it was asking permission for.

This is not the same as the more familiar Gatekeeper warning about unidentified developers, which is about *who signed* an app. Like essentially every professional Final Cut and Logic plugin, PowerDuck is distributed directly rather than through the Mac App Store, and it is signed with an Apple Developer ID and notarized by Apple on every release, so it installs and runs normally under macOS's standard security

setting, the one that allows software from identified developers. You do not need to change that setting, and you should not set it to App Store only: that blocks all directly distributed software, which is to say every audio plugin you own.

If you declined the Final Cut prompt, nothing on your Mac was changed. If you accepted it and want to clear it, quit Final Cut, then hold Option and Command while launching it and choose Delete Preferences (this also resets other Final Cut preferences, though your libraries and projects are untouched).

Troubleshooting

"The waveform on the timeline didn't change." That's Final Cut's drawing cache, not the ducking. See Timeline waveforms. Judge by ear and the Duck History strip, or Bake to keyframes to make it visible.

- **No ducking?** Check the chips on the Music/SFX copy: *"Waiting for Voice..."* means no Voice copy is feeding that Link Group: confirm the dialogue clip has PowerDuck in **Voice** mode on the **same group**, then play. The link comes alive during playback (or export). Seeing this on multicam or synchronized clips? Update to 0.5.2, which fixed exactly that.
- **Plugin not in the effects list?** Restart Final Cut Pro: it scans at launch.
- **Ducking inconsistent across many individual clips?** Compound them: one Voice on a compound clip hears the whole conversation as one continuous piece and is the most dependable setup by far.
- **"Couldn't reach Payhip" when activating, but you're online?** Something on the Mac (firewall app, VPN, DNS filter) is blocking the plugin's connection. Use the **Activate in your browser instead** link that appears right below the Activate button: the browser does the online part and gives you a code that works offline.
- **Auto (Detect) did nothing when you dropped it?** With Final Cut's Background Render off, the verdict lands on the first play instead of on drop. Or skip detection entirely: click Voice or Music yourself.
- **Ducks on room tone / background noise?** Raise **Voice Threshold** until the strip's yellow ticks only light on speech.
- **Misses quiet speakers?** Lower Voice Threshold, or better, level the dialogue with PowerEQ first.
- **Music pumps between words?** Raise **Bridge Gaps**: pauses shorter than that window hold the duck rock-steady. Lengthening **Fade Up** softens what is left. (**Hold** defaults to 0 now: Bridge Gaps does this job from the learned timeline, and a fixed hold on top only delayed the recovery at true endings.)
- **Duck feels late on the first play after opening a project?** Play through once: the link warms up on the first pass and is instant after that.
- **Timeline waveform doesn't show the duck:** expected: see the history strip.
- Have an idea or a feature wish? **Setup → Send Feedback** opens a pre-addressed email straight to the developer. Several of PowerDuck's features started as user emails.

- Something crashed or misbehaved? Email apps@rafaelludwig.com and attach the newest file starting with your editor's name from ~/Library/Logs/DiagnosticReports if there was a crash.

Uninstall

One click: download the PowerDuck Uninstaller, open it and follow the prompts. It removes both plugin files and then asks whether to keep your license and learned timeline data. Keep them if you might reinstall, since your license activation lives there. (Signed & notarized, like everything else here.)

Prefer to do it by hand? The installer places exactly two files on your Mac. To remove PowerDuck:

1. Quit Final Cut Pro (and any other host).
2. In Finder choose *Go → Go to Folder…*, paste /Library/Audio/Plug-Ins/Components and move **PowerDuck.component** to the Trash (macOS asks for your admin password).
3. Do the same in /Library/Audio/Plug-Ins/VST3 for **PowerDuck.vst3**.
4. Restart the editor.

Your license lives in ~/Library/Application Support/PowerDuck: leave it and a future re-install activates instantly. Timelines that used PowerDuck show a "missing effect" notice after removal; the rest of the project is unaffected.