E Echograph

PAID PLUGIN

Hider — a quick guide

VST3 / AU — encode and decode



So you've got Hider installed. Hooray!

What does Hider do?

Hider takes a second audio signal (the "hidden" one) and tucks it up above 20kHz. Nobody hears it in regular playback. Run that same file through Hider's decode mode (or the free **Hider Decoder**) and it comes back out. Think of it as a watermark you can actually listen to or a track hidden for your listeners who know about it.

One important part of Hider is that it requires high-sample rate files to work properly. CD quality (44.1kHz) is… ok. Not great bandwidth for the hidden signal but it would be there if you wanted. We recommend 96kHz+.

This also means that a track hidden by hider probably wont' show up on streaming services since they not only lower sample rates but also use heavy compression in order to lower your data usage. Therefore, we

recommend allowing your listeners to download a full, uncompressed, high sample rate version of your Hider track to decode on their own computer.

Before you open it: the routing thing

Hider needs *four channels*, two for your main signal, two for the thing you're hiding. It's a sidechain input, the same mechanism you'd use to duck a track off a kick drum.

- **REAPER** — just works. Route your hidden source into the sidechain input like you would for any other sidechain plugin.
- **Logic** — look for the "Side Chain" dropdown in the plugin header. It'll be grayed out only if nothing's producing sound yet on the source you want — Logic hides silent sources from the list, which looks like a bug but isn't.
- **Ableton Live** — works great on **Live 12+**. If you're on Live 11 or earlier, it may not expose the sidechain input at all.
- **Cubase / Nuendo** — the gear icon on the plugin, then "Add Side-Chain Source."
- **Studio One** — supported, should show up as a sidechain option on the insert.
- **Adobe Audition** — supported through its general VST3 sidechain handling.

Setting it up

Load Hider on the track carrying your main signal, with your hidden source routed into the sidechain input (see above). You'll land on the encode face first.

Two knobs feed the mix: **MAIN** trims the level coming in on your main input, **HIDDEN** trims the level of the thing you're hiding before it gets modulated up above 20kHz. Get the HIDDEN level too hot and you'll push artifacts down into the audible band when it's decoded; too quiet and it won't survive whatever lossy processing happens between you and your listener. Watch the meters, not your ears — you're not supposed to hear this signal at normal listening levels, so the meters are the only real feedback you've got.

OUTPUT is the final trim on the combined mix, after the hidden signal's been folded in. That's the one your listeners actually hear.

There's a **Listen to Hidden** switch that temporarily routes the hidden input straight to monitoring, bypassing the modulation — useful for confirming you've routed the right source before you commit to an encode, since otherwise you're mixing blind.

Picking a carrier frequency

Next to the meters is the carrier readout — the frequency the hidden signal rides on. Hider snaps to a few preset carriers depending on your sample rate, selectable from **CARRIER PRESETS**, or you can turn **TUNE** to dial in a custom one on a 1kHz grid. Higher sample rates buy you more carrier headroom — more bandwidth to push the hidden signal into. If you're not sure, leave it on the default; it's picked to be safe for your rate.

Whatever carrier you land on, write it down. Your listener needs that exact number to decode later — more on that below.

Decoding what you made

Flip to decode mode (or hand the file to someone running the free **Hider Decoder**) and the hidden signal comes back out on its own. The only thing you would need to supply a listener with is the carrier frequency which was used for encoding. There's no secret handshake, no key exchange. If it was encoded, it decodes.

If something sounds off

Hidden signal comes back as silence or noise. Almost always a carrier mismatch — the decode side needs the exact carrier frequency the encode side used. Double check the number.

Hidden signal doesn't survive at all, even right after encoding. Check your sample rate before anything else. This whole thing lives above 20kHz, and if your session (or whatever it gets bounced down to) is running at 44.1kHz, there's barely any room up there to begin with — and the first lossy export, stream, or re-encode along the way will likely wipe it out. Work at 96kHz or higher, and keep the file uncompressed for as long as you can before it reaches your listener.

It worked in my DAW but not after I uploaded it somewhere. That's the streaming-service problem from above, not a bug — they resample and compress, and the hidden signal doesn't survive either. Hand listeners the real file.

The fine print

14-day refund, no questions asked. Full terms in the EULA that shipped with your install. No telemetry — this thing doesn’t phone home, ever.