

TakeTown Audio — Install & Setup Guide

Oct 7, 2026 · @Christopher Conn

Everything you need to install the TakeTown suite and set up the AI Link on Windows: third-party software, the installer, Studio One, the Hub, Claude and REAPER. For what each plug-in does, see the TakeTown Audio — Plugin Guide.

1. What's in the suite

TakeTown has four layers. Only the first is required.

1.1 The plug-ins (VST3, required)

28 VST3 plug-ins. In your DAW they appear under the vendor **TakeTown Audio**.

Plug-in	What it is
AI Mastering	The full mastering chain in one plug-in, with Auto Master (a measurement-based, rule-based solver, not a neural network), A/B, macros and presets
TT-EQ	A free-band dynamic EQ with a sidechain
TT-Harmonics	Analogue-style saturation focused on a band
TT-Multiband	Up to four-band compression and expansion
TT-Comp	A compressor with five circuit characters
TT-Delay	A tempo-synced tape delay
TT-Reverb	Five reverb spaces
TT-DeEsser	A de-esser
TT-Resonance	Finds and tames ringing, harsh frequencies as they happen
TT-Repair	Cleans up the top end of AI-generated and MP3-degraded audio (AI Smooth)
TT-Limiter	A true-peak limiter with a clipper in front

Plug-in	What it is
TT-Space	Carves room in one track for another, from a sidechain key
TT-Stereo	Mono-safe widening
TT-Transient	Attack and sustain shaping
TT-Modulation	Chorus, flanger, phaser, doubler, tremolo and more, plus Natural Double
TT-Analyzer	Measures and reports (loudness, key, tempo, spectrum), changes nothing
TT-Gate	A gate, expander or ducker with a sidechain
TT-Tune	Pitch correction for a single voice; also a note editor (ARA) in Studio One
TT-Pre	A channel input stage with Auto Gain Stage
TT-Vox	A one-knob vocal channel (Simple), and from 2.0 the whole vocal suite in one window (Pro)
TT-Match	Reference matching
TT-Clip	A clipper
TT-Rider	An automatic vocal level rider
TT-Morph	A voice changer (pitch and formants), with a MIDI input
TT-Link	Changes nothing. It puts a track with no other TakeTown plug-in on the AI Link. Free, never restricted
TT-Pump	Tempo-synced volume shaping: the sidechain pump without a sidechain
TT-Wave	A polyphonic wavetable synthesizer, the suite's first instrument
TT-Compose	An instrument that plays the MIDI parts Claude composes, in sync with your song

The plug-ins work in any Windows VST3 host on their own, with no hub, no Claude and no internet connection.

1.2 The TakeTown Hub, also called the AI Link (optional)

The hub is a small program that runs in the Windows notification area (system tray). It has an orange "TT" icon. Every TakeTown plug-in reports to it: what track it is on, its settings and what it measures. The hub also reads your saved Studio One song file. Claude talks to the hub through `taketown-mcp.exe`, an MCP server installed with the hub.

You do not start the hub yourself. The first TakeTown plug-in that loads starts it, and so does Claude when it calls a TakeTown tool. The hub quits by itself about 3 minutes after the last plug-in closes.

1.3 Studio One integration (optional, recommended for Studio One users)

- **TakeTown Link:** a control-surface device for Studio One that the hub drives. It gives Claude exact access to the mixer (faders, pans, sends, mute, solo), inserts (any maker's plug-ins: add, remove, bypass, parameters), tracks, buses, routing, instruments and sound design, MIDI parts written into the song, the playhead and the tempo.
- **Mackie Control fallback:** Studio One's own Mackie Control device, pointed at the hub. The hub falls back to it for transport and mixer moves when TakeTown Link is not connected.
- **The TakeTown Compose Tasks extension:** adds **Insert Composed Part** to Studio One's Musical Functions, so Claude can write composed MIDI straight into a part.

Both devices talk to the hub through **loopMIDI** virtual MIDI ports, a free third-party program you install yourself (section 3.1).

1.4 The REAPER bridge (optional, for REAPER users)

A script REAPER runs in the background that gives the hub the same kind of control in REAPER. It needs no MIDI ports. It is new and has not yet been tried in a live REAPER session (see section 8).

1.5 Claude integration (optional)

Registers the TakeTown MCP server with **Claude Code** and **Claude Desktop**, and installs three Claude Code agents and two skills:

Agent	What it does
<code>taketown-mixer</code>	Mix engineer in act mode: measures, then makes the changes and tells you how to undo each one
<code>taketown-advisor</code>	Mix engineer in discuss mode: measures, explains, suggests a plan; it cannot change your song
<code>taketown-producer</code>	Music producer: writes beats, bass lines, chords, melodies and whole arrangements into your song

The skills (`taketown-mix` , `taketown-compose`) are the agents' handbooks. You don't call them yourself.

1.6 Optional extras

- **Audio to MIDI:** a Python helper (about 1.8 GB) for `audio_to_midi` (drums, bass, piano, melody to `.mid`), `detect_key` , and `smooth_vocal` 's offline render. It is a **separate download**, `TakeTown-AudioToMidi-Setup-<version>.exe` , that you run after the main installer (section 6.6).
- **MIDI library index:** if you own MIDI packs, the hub can index them so the producer agent can build beats and chord parts from them. Nothing is copied or shipped.

1.7 What's required and what's optional

You want	You need
The plug-ins in any DAW	The installer (plug-ins component)
Claude to see and change TakeTown plug-in settings, listen to tracks, compose MIDI files	+ the hub, + Claude Code or Claude Desktop, + the Claude integration
Claude to work the Studio One mixer, inserts, tracks, playhead, tempo, and write MIDI into the song	+ Studio One 6 (7 and Studio Pro: installed, untested), + loopMIDI with the TakeTown Link ports, + the Studio One integration
The same in REAPER	+ REAPER, + the REAPER bridge
Audio to MIDI, key detection	+ the hub, + the Audio to MIDI download (<code>TakeTown-AudioToMidi-Setup</code> , which brings uv)

2. System requirements

2.1 Computer

- **Windows 10 or Windows 11, 64-bit (x64).** The installer only runs on x64.
- **Disk space** (approximate, from the installer's manifest): see the table below.
- **Internet** is needed only to download the software, and for Claude itself. The plug-ins and the hub never send anything over the network.

Part	Size
Plug-ins	about 150 MB
Hub and integrations	about 45 MB
Audio to MIDI (optional, separate download)	about 1.8 GB: the Python environment (about 1.6 GB, most of it PyTorch) and the piano model (165 MB). The download's temporary cache (about the same again) is deleted when its setup succeeds; set up by hand, uv keeps its cache until <code>uv cache clean</code>
Vocal smoothing's own environment (optional, rarely needed)	about 250 MB

2.2 DAWs

- **Any Windows VST3 host** runs the plug-ins: Studio One, REAPER, Cubase, Ableton Live, Bitwig Studio, FL Studio, Cakewalk and others.
- **Studio One 6** gets the full AI integration (TakeTown Link, Mackie fallback, Compose Tasks). It was developed and tested on Studio One 6.6.4.
- **Studio One 7 and Fender Studio Pro 8** (Studio One's new name): the installer finds them (`%APPDATA%\PreSonus\Studio One 7` , `%APPDATA%\Fender\Studio Pro 8`) and installs TakeTown Link and the Compose Tasks extension into **every** Studio One it finds, 6 included. **They have not been tested live:** the plug-ins and the plug-in-level AI tools work there as in any VST3 host, but whether TakeTown Link and the extension run there as in Studio One 6 is not known yet. `install_status` says so when it finds them.
- **REAPER** gets near-full AI integration through the REAPER bridge (not yet tested live).

- **TT-Tune's note editor** needs an ARA host (Studio One; also Cubase, REAPER and Cakewalk).

2.3 What works in which DAW

Feature	Studio One 6	REAPER (with bridge)	Other DAWs
The plug-ins themselves	yes	yes	yes
Claude reads and sets TakeTown plug-in settings, presets and buttons	yes	yes	yes
listen (live measurements), track roles, aliases, notes, feedback	yes	yes	yes
Snapshots of TakeTown settings, undo of Claude's changes	yes	yes	yes
Compose MIDI, build beats, chords from your library (as .mid files)	yes, beside the song	yes	yes, from a tempo you give (files you drag in)
Audio to MIDI, key detection, vocal smoothing and tuning	from the saved song's tracks	from files	from files you name
Project map with other makers' plug-ins and routing	yes (from the saved song and live)	yes (live)	no
Faders, pan, mute, solo, sends	yes	yes	no (Claude gives you exact instructions instead)
Insert, remove, bypass any plug-in; any plug-in's parameters	yes	yes	no

Feature	Studio One 6	REAPER (with bridge)	Other DAWs
Add, rename, remove, colour, duplicate tracks; buses and routing	yes (buses and routing need TakeTown Link)	tracks yes; buses and routing not yet	no
Play, stop, locate, markers, tempo	yes	yes	play/stop only through a Mackie Control device you set up (untested)
MIDI parts written straight into the song	yes	yes	no
Sound design on instruments (Mai Tai, Serum, Vital, Surge XT, TT-Wave...)	yes	yes	no

3. Third-party software

Install these **before** running the TakeTown installer. The installer looks for them and ticks the matching integrations for you.

3.1 loopMIDI (Studio One users)

loopMIDI is a free virtual MIDI cable program by Tobias Erichsen. TakeTown does not include it and does not install a MIDI driver of its own. The hub opens loopMIDI's ports by name, so **the names must be exact**: same spelling, same capitals, same spaces.

Download: [loopMIDI — official page](#) (take the current version from there).

Install:

1. Download the installer from the page above and run it. Accept the defaults.
2. Start **loopMIDI** from the Start menu. Its window lists the ports, with a text box for a new port's name and a + button under the list.

Create the ports. For each name below: type it into the new-port-name box exactly as shown, then click +.

Port name (type exactly)	Needed for	Required?
TakeTown Link to S1	TakeTown Link (the hub writes, Studio One reads)	Yes
TakeTown Link from S1	TakeTown Link (Studio One writes, the hub reads)	Yes
TakeTown MCU to S1	Mackie Control fallback	Recommended
TakeTown MCU from S1	Mackie Control fallback	Recommended

Start with Windows. Right-click loopMIDI's icon in the notification area and turn on autostart. Closing loopMIDI's window only minimises it to the tray. The ports exist only while loopMIDI is running, and **loopMIDI must be running before Studio One starts.**

What you should see: four lines in loopMIDI's list with exactly the names above. While Studio One and the hub are connected, the byte counters next to the Link ports keep moving. `install_status` checks all four: the two Link ports are required, the two MCU ports are recommended (missing ones are reported as `partial` with a "Recommended:" fix, not as a failure).

If something is off:

- A typo in a name: select the port, click - to delete it, and add it again.
- Ports exist, but the counters stay at 0 after Studio One is set up: restart Windows. This was seen live: ports created while the Windows MIDI Service was running were not wired up until a reboot.

3.2 Claude Code and/or Claude Desktop (for the AI features)

You need at least one. Claude Code is the main way to use TakeTown, because the three TakeTown agents are Claude Code agents.

Claude Code (a command-line app in a terminal window): [official page](#) · [setup documentation](#). It needs a paid Claude plan (Pro, Max, Team or Enterprise) or a Claude Console account. The free plan does not include it.

1. Open **PowerShell**. Press Start, type `PowerShell`, and press Enter.
2. Run the official installer: `irm https://claude.ai/install.ps1 | iex` (or with WinGet: `winget install Anthropic.ClaudeCode`).

3. Close PowerShell, open a new one, and run `claude --version`. A version number means it worked.
4. Run `claude`. On first start it opens your browser to sign in. Follow the prompts.
5. Type `/exit` to leave it for now. (Git for Windows is optional for Claude Code. TakeTown does not need it.)

Claude Desktop (the desktop chat app): [official download](#); pick the Windows version.

1. Install it and sign in.
2. Open **Settings > Developer > Edit Config** once. This creates Claude Desktop's config file (`%APPDATA%\Claude\claude_desktop_config.json`). The TakeTown installer looks for that file to tell that Claude Desktop is there.
3. Quit Claude Desktop before running the TakeTown installer.

How the TakeTown MCP server is registered. You normally do nothing: the installer's **Claude integration** component does it (section 4). For reference, or to do it by hand:

- Claude Code (in PowerShell): `claude mcp add taketown --scope user -- "$env:LOCALAPPDATA\TakeTown Audio\Hub\taketown-mcp.exe"`. In a Command Prompt, write the path as `"%LOCALAPPDATA%\TakeTown Audio\Hub\taketown-mcp.exe"`. Then `claude mcp list` should list `taketown` as connected.
- Claude Desktop: **Settings > Developer > Edit Config**, add a `"taketown"` entry under `"mcpServers"` in `claude_desktop_config.json` with `"command": "C:\\Users\\<you>\\AppData\\Local\\TakeTown Audio\\Hub\\taketown-mcp.exe"` and `"args": []` (replace `<you>` with your Windows user name), save, and restart Claude Desktop. If the file already has an `mcpServers` section, add only the `taketown` entry inside it.

Section 7 shows how to check that Claude sees the tools.

3.3 uv (only for Audio to MIDI)

uv is a Python package manager from Astral. TakeTown's Audio to MIDI helper uses it to make its own private Python environment. You do not install Python yourself: uv downloads Python 3.10 on its own if needed. Nothing is installed system-wide.

With the Audio to MIDI download you do not need to install uv: `TakeTown-AudioToMidi-Setup` brings its own copy (uv 0.12.23, MIT or Apache-2.0, its licences installed beside it) and uses it only for this. Install uv yourself only to set Audio to MIDI up by hand ([official documentation](#)):

1. In PowerShell, run `winget install --id=astral-sh.uv -e` (the official alternative: `powershell -ExecutionPolicy Bypass -c "irm https://astral.sh/uv/install.ps1 | iex"`).
2. Close PowerShell and open a new one. `uv --version` should print a version number.

What the TakeTown setup script downloads with it (`audio2midi\setup.ps1` , run by the Audio to MIDI download, or by you): Python 3.10 if this computer does not have it; the Python packages for transcription (PyTorch CPU alone is about 1.1 GB); the piano transcription model (165 MB, CC BY 4.0, from Zenodo record 4034264). Everything goes into `%LOCALAPPDATA%\TakeTown Audio\Hub\audio2midi\` . Running the script again keeps what is already there.

Vocal smoothing (`smooth_vocal`) uses the Audio to MIDI environment, so it needs no setup of its own. `vocalsmooth\setup.ps1` in the hub folder makes it a separate environment (about 250 MB), but only if you want that. Per-note vocal tuning (`analyze_vocal_notes` , `tune_vocal`) needs no Python at all.

3.4 REAPER (only if you use it)

[REAPER — official site](#): download the current Windows x64 version from its Download page. Install REAPER and start it once before running the TakeTown installer, so its settings folder (`%APPDATA%\REAPER`) exists. The installer then finds it.

4. Running the TakeTown installer

4.1 Before you start

1. Install the third-party software you need (section 3).
2. **Close every DAW** (Studio One, REAPER, ...). The installer cannot replace a plug-in a DAW has loaded. If one is still open, the installer offers to close it.
3. Close Claude Desktop if it is open.

4.2 Run it

1. Double-click `TakeTown-Setup-<version>.exe` (for the beta: `TakeTown-Setup-0.9.0-beta.exe`).
2. **SmartScreen**. The installer is not code-signed yet, so Windows may show "Windows protected your PC". Click **More info**, then **Run anyway**. Only do this for an installer you got from TakeTown (www.Cloud9TN.com).

3. There is **no administrator prompt** at the start. TakeTown installs for your Windows user only.
4. The **Select Components** page opens. The text at the top says what was found on this computer (each integration with "(yes)" or "(no)"). The integrations are ticked where their DAW or Claude was found.

4.3 The components page: what each box does

Component	Ticked by default	Tick it when	What it installs
TakeTown plug-ins (VST3)	Always (cannot be unticked)	Always. You can untick single plug-ins in its sub-list	One <code>.vst3</code> per plug-in into the folder you choose on the next page (below; <code>%LOCALAPPDATA%\Programs\Common\VST3</code> by default)
TakeTown Hub (AI Link)	Yes	You want any AI feature	The hub, the MCP server, <code>vocaltune.exe</code> , <code>vst3params.exe</code> , the instrument books and the integration scripts, into <code>%LOCALAPPDATA%\TakeTown Audio\Hub</code>
Studio One integration	When Studio One is found	You use Studio One 6 (7 and Studio Pro 8: installed too, untested)	The TakeTown Link device and the Compose Tasks extension, into the per-user settings folder of every Studio One found
REAPER bridge	When REAPER is found	You use REAPER	One script in REAPER's Scripts folder, registered to start with REAPER
Virtual MIDI ports (loopMIDI)	When loopMIDI is found	Leave as it is	Nothing. It is a reminder: loopMIDI is installed by you (section 3.1)
Claude integration	When Claude Code or Claude Desktop is found	You use Claude	Registers the <code>taketown</code> MCP server with Claude Code and Claude Desktop; installs the agents and skills into <code>%USERPROFILE%\Claude</code>

Component	Ticked by default	Tick it when	What it installs
MIDI library index	Yes	Leave it ticked	Nothing now. The hub indexes your own MIDI packs later, when you ask (section 6.5)

The integrations need the hub. If you untick the hub, they are unticked too.

Audio to MIDI is not on this page: it is a separate download (section 6.6), because it adds about 1.8 GB. The hub still installs its scripts, so the download only adds the environment.

Next: "Where should the plug-ins go?" Pick the VST3 folder your DAW scans:

Option	Pick it when	Administrator rights
Your user's VST3 folder (%LOCALAPPDATA%\Programs\Common\VST3 , the default)	Studio One or REAPER with this folder in their plug-in paths (5.1; REAPER: Preferences > Plug-ins > VST)	None
All users' VST3 folder (C:\Program Files\Common Files\VST3)	You use Ableton Live, or any DAW that does not find the per-user folder: every VST3 host scans this one	Windows asks once, when you click Install
A folder your DAW already scans (listed only when found: from Studio One's VST plug-in locations, Studio Pro's, or REAPER's VST path, labelled "scanned by ...")	You keep your plug-ins in your own folder	Only if your user cannot write there
Another folder (type it or Browse...)	Anywhere else; the installer makes it if needed	Only if your user cannot write there

If Ableton Live is found, the per-user option says so: Live scans only the all-users folder and its one VST3 custom folder. The installer remembers the folder: the next run (an update) offers the same one, and moving to another removes the old copy. If you decline

the administrator prompt, you stay on the Ready page: click **Back** and pick your user's folder, or **Install** to be asked again. Everything else the installer does stays per user.

4.4 The one admin prompt (optional task)

With the Studio One integration ticked, the next page offers one task, **off by default**:

Also put the Compose Tasks extension in Studio One's program Scripts folder (asks for administrator rights; only if Studio One does not list Insert Composed Part otherwise)

Leave it off the first time. Only turn it on (by running the installer again) if section 5.5's check fails. When it is on, Windows asks once for administrator rights (UAC) at the end of the install. It writes one file per Studio One program folder found: `C:\Program Files\PreSonus\Studio One 6\Scripts\TakeTownComposeTasks.package` (and the same under Studio One 7, or `C:\Program Files\Fender\Studio Pro 8`, when those are installed). The uninstaller finds these files and offers to remove them (section 13.2).

4.5 The finish page

The finish page lists your next steps for exactly what you installed, for example:

- rescan plug-ins in your DAW if they do not show;
- Studio One: restart it, then add the TakeTown Link device with its ports;
- loopMIDI: the exact port names;
- REAPER: restart it;
- Claude: run `/mcp` in Claude Code, restart Claude Desktop;
- Audio to MIDI: the name of the separate download and where to get it (section 6.6);
- you can ask the agent for `install_status` at any time.

The rest of this guide walks through each of these.

4.6 Install locations

Everything is per user, the Audio to MIDI download too. Nothing goes into Program Files except the plug-ins when you choose the all-users folder (4.3), and the optional task above.

What	Where
Plug-ins	The folder chosen in 4.3: by default <code>%LOCALAPPDATA%\Programs\Common\VST3</code> (the per-user VST3 folder), or <code>C:\Program Files\Common Files\VST3</code> (all users), or your own. Recorded in <code>HKCU\Software\TakeTown Audio\TakeTown (Vst3Dir)</code> and the hub's <code>locations.json</code> ; <code>install_status</code> says where they are
Hub, MCP server, scripts, your hub data	<code>%LOCALAPPDATA%\TakeTown Audio\Hub</code>
TakeTown Link device	<code>%APPDATA%\PreSonus\Studio One 6\User Devices\TakeTown\Link</code> (and the same in every other Studio One folder found: <code>Studio One 7</code> , <code>%APPDATA%\Fender\Studio Pro 8</code>)
Compose Tasks extension	<code>%APPDATA%\PreSonus\Studio One 6\Extensions\TakeTownComposeTasks</code> (likewise)
Audio to MIDI download	<code>%LOCALAPPDATA%\TakeTown Audio\AudioToMidi</code> (uv, its Python, the setup log); the environment and model in the hub's <code>audio2midi</code> folder
REAPER bridge	<code>%APPDATA%\REAPER\Scripts\TakeTown\taketown_bridge.lua</code> (plus a block in <code>Scripts__startup.lua</code>)
Claude agents and skills	<code>%USERPROFILE%\\.claude\agents</code> , <code>%USERPROFILE%\\.claude\skills</code>
Licence keys	<code>%APPDATA%\TakeTown Audio\licence.key</code>

`%LOCALAPPDATA%` is usually `C:\Users\<you>\AppData\Local`, and `%APPDATA%` is `C:\Users\<you>\AppData\Roaming`. Paste either into File Explorer's address bar to open it.

4.7 Installed a DAW or Claude later? Run the installer again

The components page is ticked from what is on the computer **every time it runs**. Install the new DAW (or Claude), start it once, then run the TakeTown installer again. The new integration is ticked for you. Everything else is replaced in place, and your data is kept.

5. Studio One setup

Do these once, in order. Before you start, loopMIDI must be running with the TakeTown ports (section 3.1).

5.1 Find the plug-ins

1. Start Studio One.
2. Open the **Browser** (F5) and look under **Effects** and **Instruments**. The TakeTown plug-ins are listed under the vendor **TakeTown Audio** (TT-Wave and TT-Compose are instruments, the rest are effects).
3. **If they are not there, rescan:**
 - a. **Studio One > Options > Locations > VST Plug-Ins.**
 - b. Check that the folder you chose in the installer (4.3) is in the list: for the default, `C:\Users\<you>\AppData\Local\Programs\Common\VST3` . If not, click **Add**, pick that folder, and confirm. (The all-users folder, `C:\Program Files\Common Files\VST3` , Studio One scans without being listed. A folder the installer offered as "scanned by Studio One" is already in this list.) Not sure which folder? Ask the agent for `install_status` : the plug-ins' where .
 - c. Click **Reset Blacklist**, then **Rescan** (or restart Studio One).

Worked: the TakeTown plug-ins are in the Browser, and one dropped on a track opens its window with a licence line at the bottom.

5.2 Add the TakeTown Link device

The installer copied the device into Studio One's `User Devices` folder. Studio One only loads devices when it starts, so **restart Studio One after installing**.

1. **Studio One > Options > External Devices.**
2. Click **Add....**
3. In the list on the left, open **TakeTown** and choose **Link**.
4. Set **Receive From:** `TakeTown Link to S1` and **Send To:** `TakeTown Link from S1` .
Leave MIDI clock and timecode sending off.
5. Click **OK**, then **OK** again to close Options.

The names describe the hub's direction: "to S1" carries the hub's messages to Studio One, so Studio One receives from it. **Swapping the two is the most common mistake.**

Worked: in loopMIDI the byte counters on both Link ports move (the device sends a heartbeat every 2 seconds). Claude's `transport` status shows the Link as connected (section 10).

If TakeTown > Link is not in the Add list:

- Studio One was not restarted after the install: restart it.
- Check that the device's six files are in `%APPDATA%\PreSonus\Studio One 6\User Devices\TakeTown\Link\` (Studio One 7: `...\Studio One 7\...`; Studio Pro 8: `%APPDATA%\Fender\Studio Pro 8\User Devices\...`). If not, run the installer again with Studio One integration ticked, or run (PowerShell; it installs into every Studio One found, and `-List` shows which): `powershell -ExecutionPolicy Bypass -File "$env:LOCALAPPDATA\TakeTown Audio\Hub\integrations\s1-surface\install.ps1"`
- Last resort: copy the same six files, as administrator, to `C:\Program Files\PreSonus\Studio One 6\devices\TakeTown\Link\` (where Studio One's own devices live; a Studio One update may remove them).

5.3 Add the Mackie Control fallback (recommended)

The hub prefers TakeTown Link. When the Link is not connected, it uses this Mackie Control device for play, stop, rewind, fast forward, loop, markers and mixer moves. Rewind, fast forward and loop always go through Mackie.

1. In loopMIDI, make sure `TakeTown MCU to S1` and `TakeTown MCU from S1` exist.
2. **Studio One > Options > External Devices > Add...**
3. Open **Mackie** and choose **Control**.
4. Set **Receive From:** `TakeTown MCU to S1` and **Send To:** `TakeTown MCU from S1`. Leave MIDI clock and timecode sending off.
5. **OK, OK.**

Worked: within a few seconds the hub opens the ports. Claude's `transport` status shows `connected: true` for the Mackie surface once Studio One has sent it anything. Pressing play once helps.

Both devices can be set up at the same time. They use separate ports.

5.4 Leave both devices as they are

The devices have no window of their own (the Edit window in External Devices is empty; that is normal). Do not assign them as your keyboard or as a MIDI input to tracks.

5.5 Check the Compose Tasks extension

This adds the **Insert Composed Part** command Claude uses to write MIDI into a part.

1. After the install, restart Studio One.
2. Open a song, select an instrument part (a MIDI event) on an instrument track.
3. Open **Event > Musical Functions**.

Worked: the list has **Insert Composed Part**. If you click it yourself, it says no part is waiting. That is expected: Claude prepares the part first.

If it is not listed:

1. Run the TakeTown installer again and tick the task **Also put the Compose Tasks extension in Studio One's program Scripts folder** (one admin prompt). Or, in an administrator PowerShell: `powershell -ExecutionPolicy Bypass -File "$env:LOCALAPPDATA\TakeTown Audio\Hub\integrations\s1-extension\install.ps1" -Target Scripts` (To open an administrator PowerShell: Start, type `PowerShell`, right-click it, **Run as administrator**. If Windows asks for a different administrator account's password, type the full path `C:\Users\<you>\AppData\Local\TakeTown Audio\Hub\...` instead of `$env:LOCALAPPDATA`, because that account has its own folder.)
2. Restart Studio One and check again.

5.6 Restart Studio One after every TakeTown update

Studio One loads device scripts and extensions **only when it starts**. After any install or update of TakeTown, restart Studio One once. Until you do, the hub notices that Studio One is running an older TakeTown Link than the one installed, says "restart Studio One", and the tools that need the new version refuse.

Studio One can be restarted at any time, but **save your song first**. The agent never saves for you.

5.7 Every session

- Start loopMIDI first (automatic if autostart is on), then Studio One.
- Put at least one TakeTown plug-in in the song. That starts the hub and puts the song on the AI Link. On tracks with no other TakeTown plug-in, TT-Link reports the track and changes nothing.

- **Save the song (Ctrl+S)** before asking Claude about other makers' plug-ins, routing or audio clips. The hub reads those from the saved file.

6. The TakeTown Hub

6.1 What it is and how it starts

`TakeTownHub.exe` in `%LOCALAPPDATA%\TakeTown Audio\Hub`. It:

- collects what every TakeTown plug-in reports;
- reads your saved Studio One songs (read-only: it never writes a song);
- drives TakeTown Link, the Mackie Control fallback and the REAPER bridge;
- answers Claude's tool calls (through `taketown-mcp.exe`).

It starts by itself when a TakeTown plug-in loads, or when Claude calls a TakeTown tool. It quits 3 minutes after the last plug-in closes. Only one hub runs per Windows user.

Everything stays on your computer: the plug-ins and Claude reach the hub through local named pipes that only your user can open. Nothing listens on the network, and no audio is uploaded.

6.2 The tray menu

Click the orange **TT** icon in the notification area (you may need to click the **^** arrow to see it):

Item	Does
(status line)	The active song, how many others are open
Link song...	Fallback: tells the hub which <code>.song</code> file to read when it cannot find it
Unlink song	Goes back to following the open song
Re-read song now	Reads the saved song again
Songs folders	The folders searched for songs, in order; Add folder... , Remove , Use Studio One's folders only
Open log	Opens <code>hub.log</code>
Open hub folder	Opens <code>%LOCALAPPDATA%\TakeTown Audio\Hub</code>

Item	Does
Licence	Beta days left, the AI Link's licence, Copy machine ID
Quit	Quits the hub

6.3 The log

`%LOCALAPPDATA%\TakeTown Audio\Hub\hub.log` (the previous one is `hub.log.1`; it rotates at 2 MB). Tray > **Open log** opens it. It says which song file it read and why, when TakeTown Link connects or goes quiet, and every error.

6.4 Restarting or quitting the hub

- **Quit:** tray > **Quit**, or in PowerShell: `& "$env:LOCALAPPDATA\TakeTown Audio\Hub\TakeTownHub.exe" --quit` (`--quit` asks the running hub to close and waits up to 15 seconds).
- **Start it again:** open any TakeTown plug-in, or ask Claude anything that uses a TakeTown tool. You can also double-click `TakeTownHub.exe`.
- After the hub restarts, run `/mcp` in Claude Code and reconnect `takestown` if its tools fail.

6.5 Settings

Songs folders. The hub finds Studio One songs by itself, from Studio One's own settings (where new songs go, your content folder) and its recent-songs list. If your songs live somewhere else (another drive), add the folder:

- tray > **Songs folders** > **Add folder...**, or
- ask Claude: "Add E:\AUDIO\Studio One\Songs to the hub's song folders" (`set_songs_dirs`). The agents only change hub settings when you ask.

Ask "Where does the hub look for songs?" (`get_hub_settings`) to see every folder and the file read for each open song. The setting is stored in `settings.json` in the hub folder.

The MIDI library. If you own MIDI packs (Cookin Soul, Niko Kotoulas's progressions, Cymatics, any folder of `.mid` files), the hub can index them. Only paths and what it hears in each file are stored; your packs are never copied or uploaded. Ask Claude, for example: "Index my MIDI packs in D:\Samples\MIDI and D:\Samples\Cookin Soul."

This runs `refresh_midi_library` with those folders (pack folders, each with an optional kind `cookinoul`, `nikos`, `cymatics` or `generic`). It remembers them in `midi-library\packs.json`, so later refreshes only need "refresh the MIDI library". About 5,800 files take a few seconds. The producer agent then uses `find_midi`, `build_beat` and `add_chords_from_library`. The index is `%LOCALAPPDATA%\TakeTown Audio\Hub\midi-library\index.json`.

6.6 Setting up Audio to MIDI later

Audio to MIDI (`audio_to_midi`, `detect_key`, `smooth_vocal` 's offline render) is a separate download, because it adds about 1.8 GB:

1. Install TakeTown first (the hub must be there; the download checks and says so if not).
2. Download **TakeTown-AudioToMidi-Setup-<version>.exe** (for the beta: `TakeTown-AudioToMidi-Setup-0.9.0-beta.exe`) from the same place as the TakeTown installer (www.Cloud9TN.com), and run it. SmartScreen may warn as for the main installer (4.2). No administrator prompt.
3. After it copies its files, a **console window** opens and shows each step (lines start with `==>`): the Python environment, the packages (PyTorch is a large download), the piano model. It takes several minutes. Leave the window open; it closes by itself.
4. A message says whether it worked. The log is `%LOCALAPPDATA%\TakeTown Audio\AudioToMidi\setup.log`.

If it failed (usually a dropped connection): run the download again; it keeps what was already fetched. The message and the log say what went wrong.

By hand instead (developers): install uv (section 3.3), then in a new PowerShell window:

```
powershell -ExecutionPolicy Bypass -File "$env:LOCALAPPDATA\TakeTown
Audio\Hub\audio2midi\setup.ps1"
```

Worked: ask Claude "Is audio to MIDI set up?" (`audio_to_midi_status`). It says what it found; `install_status` lists Audio to MIDI as `installed`.

6.7 Your data

These live in the hub folder and are **yours**. Updates never touch them, and uninstalling keeps them unless you say yes to the question at the end:

File or folder	Holds
<code>state.json</code>	Track notes, roles, aliases, project notes the hub keeps per song
<code>settings.json</code>	Hub settings (songs folders, port names)
<code>feedback.jsonl</code>	Feedback you logged about plug-ins (<code>log_feedback</code>)
<code>snapshots\</code>	Mix snapshots (<code>snapshot_mix</code>) for songs not saved yet; saved songs keep theirs beside the song
<code>kitmaps\</code>	Custom drum kit maps
<code>midi-library\</code>	The MIDI library index and <code>packs.json</code>
<code>renders\</code>	Files made without a song in scope
<code>instrument-params\</code> , <code>Imports\</code> , <code>reaper-bridge\</code>	Caches and hand-off folders
<code>audio2midi\.venv</code> , <code>audio2midi\models</code> , <code>vocalsmooth\.venv</code>	The Python environments

Files Claude makes for a song (composed MIDI, vocal renders) go into `<song folder>\TakeTown Renders\` (MIDI in `TakeTown Renders\MIDI`). Notes, roles and aliases are also saved inside the plug-ins' state in your song, so they travel with it.

7. Claude setup and the TakeTown agents

7.1 Check that Claude Code sees the TakeTown tools

1. Open PowerShell (any folder) and run `claude` .
2. Type `/mcp` and press Enter. **Worked:** `taketown` is listed as connected; if it shows as failed, select it and choose reconnect.
3. Ask: `Say hello to the TakeTown hub.` Claude reports that the hub is there and whether the AI Link works. The first time Claude uses a TakeTown tool, Claude Code may ask your permission; allow it.

7.2 Check Claude Desktop

1. Restart Claude Desktop after the TakeTown install.
2. In a new chat, open the tools/connectors menu in the message box and check that **taketown** is listed and on.
3. Ask "Say hello to the TakeTown hub."

Claude Desktop gets the TakeTown tools; the three agents and the skills are Claude Code features.

7.3 Starting the agents (Claude Code)

The agents are installed for your user, so they work from any folder.

Command	Mode
<code>claude --agent taketown-mixer</code>	Act: measures, makes the changes, reports each with how to undo it
<code>claude --agent taketown-advisor</code>	Discuss: answers, measures and suggests a plan; cannot change the song
<code>claude --agent taketown-producer</code>	Producer: beats, bass lines, chords, melodies, whole arrangements
<code>claude --agent taketown-mixer "the vocals sound flat and muddy"</code>	Starts with a request

7.4 Good first prompts

With Studio One open, the song saved, loopMIDI running and at least one TakeTown plug-in on a track:

- `What do I still need to install?`
- `Give me an overview of this song.`
- `Is TakeTown Link connected?`
- `Advisor: How loud is this mix, and what would you change on the master?`
- `Mixer: The vocals sound harsh and sit too far back. Fix it.`
- `Mixer: Take a snapshot called "before mastering", then master this for streaming. — then Undo that.`

- **Producer:** Make me a neo-soul beat in C# minor at 84 bpm.
- **Index my MIDI packs in** D:\Samples\MIDI.

7.5 What the agents will and won't do

- **They never save your song.** Press **Ctrl+S** yourself to keep changes, and before asking about things that come from the saved file.
- **They never record or arm tracks**, and never press Undo, Delete or Quit in Studio One.
- Their own plug-in changes are taken back with "undo" or by recalling a snapshot; structural changes (tracks, inserts, routing, parts, tempo) with **Ctrl+Z** in Studio One.
- **Structural changes only when you ask:** adding or removing tracks or plug-ins, buses, routing, tempo changes, inserting parts.
- **The mixer agent** makes mix moves without asking first, reports each with how to undo it, and snapshots the mix before its first change of a session. **The advisor** cannot change anything.
- **They leave solos off** when they finish, and don't touch your files or hub settings unless you ask.
- **Only you can:** add new sends, drag an insert to a new position (Studio One doesn't let a script move one), name folders or groups, load TT-Match references.

8. REAPER setup

The REAPER bridge was built against REAPER's documented scripting API but **has not yet been tried in a live REAPER session**. After Claude's first change, check that REAPER shows it, and report anything odd ("log feedback that...").

8.1 Install

The installer's **REAPER bridge** component does it when REAPER is found. By hand, with REAPER closed: `powershell -ExecutionPolicy Bypass -File "$env:LOCALAPPDATA\TakeTown Audio\Hub\integrations\reaper-bridge\install.ps1"` (add `-ResourcePath D:\REAPER` for a portable REAPER; REAPER shows its folder under **Options > Show REAPER resource path**). It copies `taketown_bridge.lua` into `Scripts\TakeTown\` and adds a marked block to `Scripts\__startup.lua`; nothing else in that file is touched.

8.2 Start it

Restart REAPER: the bridge starts in the background. Without a restart, run it once by hand: **Actions > Show action list > ReaScript: Load...** and pick `taketown_bridge.lua`. No loopMIDI is needed.

8.3 Check it

Ask Claude for an overview of the project. **Worked:** the answer comes from REAPER, with your tracks, FX chains and tempo, and `install_status` shows the bridge installed and running. If Claude says "needs the TakeTown REAPER bridge", restart REAPER or load the script by hand.

Every change Claude makes in REAPER is one REAPER undo step named `TakeTown: ...`; **Ctrl+Z in REAPER** undoes it. Not yet available in REAPER: buses and routing tools, and reading MIDI parts by track.

9. Other DAWs

In Cubase, Ableton Live, Bitwig Studio, FL Studio, Cakewalk and other VST3 hosts:

- **Works:** every TakeTown plug-in; Claude reading and setting TakeTown plug-in settings, presets and buttons; `listen` and live measurements (press play); track roles, aliases and notes; snapshots and undo of TakeTown settings; feedback.
- **Works from files and a tempo you give:** composing MIDI, whole songs, beats from your MIDI library and chords (say the tempo, "at 92 bpm", or press play for a moment so a TakeTown plug-in reports it); audio to MIDI and key detection on audio files you name; vocal smoothing and tuning on a file you name. The files go into `%LOCALAPPDATA%\TakeTown Audio\Hub\renders\<project>`, and Claude tells you to drag them into your DAW.
- **Doesn't work there yet:** the project map of other plug-ins and routing, faders, pan, mute, solo, sends, inserting plug-ins, tracks, markers and locate. Claude gives you exact steps to do these by hand instead.

Tracks a DAW gives no name to appear as `<plugin> (no track name, <id>)`. Tell Claude what each one is; it remembers.

10. Check everything: install_status

Ask Claude **"What do I still need to install?"** or simply `Run install_status`. It is read-only, changes nothing, and works with no song open and even without a licence. It reads the install manifest beside the hub and checks the computer.

10.1 Reading the result

- **summary:** one paragraph, for example "Found: Studio One, loopMIDI, Claude Code. Everything that applies here is installed and current."
- **hosts:** what was found on the computer (Studio One, REAPER, loopMIDI, Claude Code, Claude Desktop).
- **components:** one entry per component, each with a **state** (table below).
- **fixes:** the to-do list, each with the exact command.

State	Means	Do
<code>installed</code>	Present and current	Nothing
<code>outdated</code>	Older than this TakeTown version	Run the installer again, or the <code>fix</code> command
<code>missing</code>	Not installed	Run the <code>fix</code> command, or the installer with the component ticked
<code>partial</code>	Some parts there, some not (for loopMIDI: some ports missing; for Studio One: installed in one version's folder and not another's)	Run the <code>fix</code>
<code>restart-needed</code>	Studio One is running an older TakeTown Link than the one installed, or the hub was replaced while running	Restart Studio One (save first), or quit the hub
<code>not-set-up</code>	An optional part not set up (Audio to MIDI, the MIDI library)	Only if you want it: for Audio to MIDI the <code>fix</code> names the download (section 6.6)
<code>not-applicable</code>	Its DAW or app is not on this computer	Nothing, unless you install that DAW: then rerun the installer

A fix looks like this:

```
REAPER found but the bridge isn't installed: mixer and MIDI control in REAPER
are off
fix: powershell -ExecutionPolicy Bypass -File
"C:\Users\you\AppData\Local\TakeTown Audio\Hub\integrations\reaper-
bridge\install.ps1"
```

The Studio One integration entry also has a **live** part: whether the Link ports are open, whether the device is connected, and the device version Studio One is running against the one installed. With more than one Studio One (6, 7, Studio Pro), each part lists every version's folder and its state. The loopMIDI entry checks all four ports: the two Link ports are required (any missing are listed as **missingPorts**, and the setup is not OK until they exist); the two Mackie ports are recommended (missing ones are listed as **missingRecommendedPorts**, the entry is `partial` with `recommended: true`, and the summary mentions them without counting them as a problem).

A hub or plug-in newer than the installer that made the manifest (an earlier development build, for example) shows as `installed` with a note, never as `outdated`.

"Run the TakeTown installer again" is always a valid fix.

10.2 Other quick checks

Ask Claude	Tells you
"Say hello to the hub" (<code>hello</code>)	The hub is running; whether the AI Link works
"Is TakeTown Link connected?" (<code>transport status</code>)	Both surfaces: connected or not, which one is preferred, the device version, any "restart Studio One" warning
"Give me an overview of this song" (<code>project_overview</code>)	The song the hub found, its tracks, the host DAW
"Where does the hub look for songs?" (<code>get_hub_settings</code>)	Song folders, the file read for each song, the Link ports
"What's my licence status?" (<code>licence_status</code>)	Machine ID, beta days left, each plug-in's status, the AI Link's

11. Licensing and activation

11.1 During the beta (until 31 December 2026)

You need no key. Until 31 December 2026 (your computer's local time), every plug-in, TT-Tune's note editor and the AI Link work fully. The licence line at the bottom of each plug-in counts the days left.

11.2 From 1 January 2027

Each plug-in is licensed on its own, and so is the AI Link. Without a key that covers it:

- a **plug-in** keeps working, but its sound dips to silence for 2 seconds once a minute, with a red banner (not when bypassed); your songs and presets are never changed;
- **TT-Compose** starts no note in that 2-second window instead;
- **TT-Tune's note editor** is view-only and the takes play as sung; your edits come back as soon as a key covers TT-Tune;
- **TT-Link** is free and never restricted;
- the **AI Link** (hub, AI tools, TakeTown Link, Mackie surface) stops: every tool answers with your machine ID.

One key can cover several products, or "all" (including products released later); keys add up. A key for plug-ins does not cover the AI Link: ask for one that includes "AI Link".

11.3 Getting and adding a key

1. Click the **licence line** at the bottom of any TakeTown plug-in. The Licence panel opens.
2. Click **Copy** next to "This machine's ID" (TT-XXXX-XXXX-XXXX ; also in the hub's tray menu, **Licence > Copy machine ID**).
3. Send the ID to Cloud9 TakeTown (www.Cloud9TN.com) with what you bought.
4. Paste the key you get back and click **Add key**.

Worked: the panel says the key was accepted and lists every plug-in's status. Do it once: every plug-in and the hub share one key file, and open plug-ins pick it up within seconds. Keys are checked on your computer; nothing is sent over the internet. If you reinstall Windows or replace the system drive, the machine ID changes: send the new ID for a new key.

12. Troubleshooting

Start with `install_status` (section 10). It finds most setup problems and gives the exact fix. The hub's log (tray > **Open log**) says what happened.

12.1 Installing

Symptom	Likely cause	Fix
"Windows protected your PC" when starting the installer	The installer is not code-signed yet (SmartScreen)	More info > Run anyway , for an installer from TakeTown only
The installer asks you to close an application	A DAW has a TakeTown plug-in loaded	Save, close the DAW, continue
An integration was not ticked	Its DAW or Claude was not found (the components page says "(no)")	Install and start that app once, then run the installer again; or tick it by hand
Audio to MIDI "not set up"	It is a separate download	Run <code>TakeTown-AudioToMidi-Setup (6.6)</code>
The Audio to MIDI download says the hub is not installed, or its scripts are too old	TakeTown is not installed yet, or is an older version	Run the TakeTown installer of the same version first, then the download again
The Audio to MIDI setup did not finish	Usually a dropped connection during the large download	Run the download again (it keeps what was fetched); the log is <code>%LOCALAPPDATA%\TakeTown Audio\AudioToMidi\setup.log</code>
The admin step says it did not run	UAC was declined	Run the command it shows from an administrator PowerShell, or skip it (it is optional)

12.2 Plug-ins

Symptom	Likely cause	Fix
TakeTown plug-ins don't show in the DAW	The DAW does not scan the folder they went to (by default the per-user VST3 folder)	<code>install_status</code> says where they are (where) and warns when a DAW found here does not scan it. Studio One: Options > Locations > VST Plug-Ins , add that folder if missing, Reset Blacklist, Rescan . REAPER: Preferences > Plug-ins > VST , add it, Re-scan . Ableton Live: set Preferences > Plug-Ins > VST3 Plug-In Custom Folder to it. Or run the installer again and pick the all-users folder (<code>C:\Program Files\Common Files\VST3</code>), which every DAW scans
The installer says the plug-ins could not be copied to the folder you chose	The administrator prompt was declined or the copy failed (a DAW had them loaded)	Close the DAW and run the installer again, allowing the prompt; or accept its offer to use your user's VST3 folder
Two copies of TakeTown plug-ins in the DAW	An old copy in another folder (installed by hand, or before you changed folders)	The installer removes the old copy when the folder changes; otherwise delete the old <code>.vst3</code> folders (<code>install_status</code> shows the current one)
A plug-in dips to silence for 2 s every minute	After the beta, no key covers it	Section 11.3
Plug-ins report nothing while playing	They are deactivated in Studio One (songs can reopen with plug-ins off after a crash)	Right-click the insert > Activate , or ask the mixer agent to activate it

12.3 The hub

Symptom	Likely cause	Fix
No TT icon in the tray; Claude says the hub isn't running	No TakeTown plug-in is loaded, or the hub quit 3 minutes after the last one closed	Open a TakeTown plug-in, or call any TakeTown tool from Claude (it starts the hub). Check that <code>TakeTownHub.exe</code> is in the hub folder
Every tool says "not licensed"	After the beta, no key covers the AI Link	Section 11; <code>licence_status</code> shows the machine ID
"No track matches", or the wrong song's tracks	Several songs open in Studio One; the hub uses the song in front	Bring your song to the front, close the others, or name the song in your request
The song's map is missing or old	The song is unsaved, or saved in a folder the hub doesn't search	Ctrl+S ; add the folder (6.5); or tray > Link song...
New tools are missing after an update	Claude Code still runs the old MCP server	<code>/mcp</code> > reconnect <code>taketown</code>

12.4 loopMIDI, TakeTown Link and Mackie Control

Symptom	Likely cause	Fix
<code>install_status</code> lists missingPorts, or says loopMIDI isn't installed	The ports don't exist or are misnamed	Install loopMIDI; add the ports with the exact names (3.1); then restart Studio One
<code>install_status</code> says "Mackie Control fallback ports missing ... (recommended)"	The two MCU ports were not added	Optional: add <code>TakeTown MCU to S1</code> and <code>TakeTown MCU from S1</code> in loopMIDI and set up 5.3
A port name is almost right ("Taketown link to S1", extra space)	Typo	Delete it in loopMIDI (-) and add it again exactly

Symptom	Likely cause	Fix
TakeTown > Link isn't in External Devices' Add list	Studio One not restarted since the install, or files missing	Section 5.2, "If TakeTown > Link is not in the Add list"
Link "not connected", or "has gone quiet"	loopMIDI started after Studio One, ports swapped, device not added, or Studio One busy (a dialog open)	Check Receive From TakeTown Link to S1, Send To TakeTown Link from S1; close any dialog; restart Studio One with loopMIDI running; try again after a few seconds
"Studio One is running TakeTown Link x.y.z; ... needs a.b.c. Restart Studio One", or restart-needed	Studio One still runs the device from before the update (it loads devices only at startup)	Save, restart Studio One
The device is listed but loopMIDI counters stay at 0	Windows MIDI Service missed ports created while it ran	Restart Windows
Mackie Control: transport says "not connected" or "Studio One has not talked to the surface"	Device missing, ports swapped, or loopMIDI started after Studio One	Receive From TakeTown MCU to S1, Send To TakeTown MCU from S1; in External Devices edit the device and re-select the ports; restart Studio One; press play once
Mackie connected but hostSeen: false right after a hub update	Studio One hasn't sent anything since the hub restarted	Press play or touch a fader, wait a few seconds
Mixer moves refused "automation"	The track's automation is in Write, Touch or Latch	Set it to Read or Off, or tell Claude you want it recorded
A Studio One "Scripting Error" dialog naming TakeTownLinkDevice.js	A device problem	Click OK; reinstall the device (5.2) and restart Studio One; report it ("log feedback...")

12.5 Playback, playhead and tempo

Symptom	Likely cause	Fix
<code>locate</code> or <code>set_tempo</code> says the playhead did not reach the bar, or "close the console (F3) or editor (F2)"	The console (F3) or the editor (F2) is open at the bottom of Studio One and hides the transport bar, or another window has the focus	Close the console or editor so the transport bar shows, click in the arrange view, and ask again
<code>locate</code> to a marker fails or stops early	Markers were added since the last save	Ctrl+S , then ask again
<code>listen</code> says no measurements arrived	The song isn't playing there, or that section is silent	Play a section where the track plays, or let the agent start playback

12.6 Mixing and the song

Symptom	Likely cause	Fix
Only one or two tracks are audible	A solo was left on	Ask the mixer agent "clear the solos" (<code>clear_solos</code>), or click the lit Solo buttons off. The advisor cannot clear solos (it is a change)
Claude's changes are gone after reopening the song	The song was not saved; the agent never saves	Ctrl+S after changes you want to keep
Claude doesn't see a plug-in or track you just added	Other makers' plug-ins and routing come from the saved song, or the live map wasn't refreshed	Ctrl+S , then ask again
Faders, tracks or inserts behave oddly, or the hub reads an old version of the song	Two copies of Studio One are running: a second window, or a leftover <code>Studio One.exe</code> with no window after a crash	Run only one Studio One. After a crash, save, close Studio One, and in Task Manager end any <code>Studio One.exe</code> that is left; then start it again

Symptom	Likely cause	Fix
"Undo" through Claude doesn't undo a track, insert or part change	Structural changes are Studio One's	Ctrl+Z in Studio One
Insert Composed Part isn't in Event > Musical Functions	Extension not loaded	Section 5.5

12.7 Claude

Symptom	Likely cause	Fix
<code>/mcp</code> doesn't list <code>taketown</code> , or shows it failed	Not registered, or the hub folder is missing	Run the installer again with Claude integration ticked, or <code>claude mcp add (3.2)</code> ; then <code>/mcp > reconnect</code>
<code>claude --agent taketown-mixer</code> says the agent doesn't exist	The agents aren't in <code>%USERPROFILE%\claude\agents</code>	Run the installer again with Claude integration ticked, or <code>integrations\claude\install.ps1</code> in the hub folder
Claude Desktop doesn't show taketown	Claude Desktop was open during the install, or its config file didn't exist then	Restart Claude Desktop; if still missing, add the entry by hand (3.2) and restart
<code>claude</code> is not recognised	Claude Code not on PATH yet	Open a new PowerShell window; see Claude Code's setup page
<code>audio_to_midi / detect_key</code> say not set up	The Audio to MIDI download hasn't been run	Section 6.6
<code>analyze_vocal_notes / tune_vocal</code> say not set up	<code>vocaltune.exe</code> missing (an old hub)	Run the current installer

13. Updating and uninstalling

13.1 Updating

1. Save your songs and close your DAWs.

2. Run the new `TakeTown-Setup-<version>.exe` over the old install. It asks the running hub to quit (`TakeTownHub.exe --quit`), replaces the files, and keeps all your data. The plug-in folder page offers the folder you chose last time (4.3); with the all-users folder, Windows asks for administrator rights once again.
3. Restart Studio One (it loads TakeTown Link and the extension only at startup). Restart REAPER if you use the bridge.
4. In Claude Code, `/mcp > reconnect taketown` (or start a new session). Restart Claude Desktop.
5. If you use Audio to MIDI, run the new `TakeTown-AudioToMidi-Setup-<version>.exe` too (it keeps the environment and updates what changed).
6. Ask for `install_status`: everything should say `installed`.

13.2 Uninstalling

Windows Settings > Apps > Installed apps > TakeTown > Uninstall.

It removes what the installer put there: the plug-ins, the hub's programs and scripts, the TakeTown Link device and the extension (from every Studio One version's folder), the REAPER script and its `__startup.lua` block, the `taketown` MCP entries in Claude, and the agents and skills.

The plug-ins come out of the folder they were installed to. If that is the all-users folder (`C:\Program Files\Common Files\VST3`), it lists them and asks once: Yes brings **one administrator prompt**; No leaves them there and names the folders to delete by hand.

If you used the admin task (4.4), it finds the extension in Studio One's program `Scripts` folder and asks once: **Remove it too?** Yes brings **one administrator prompt** (UAC) and removes that file from every Studio One program folder; No (or declining the prompt) leaves it there, harmless without TakeTown.

At the end it asks: **Also remove your TakeTown data?** The default is **No**. With No, your notes, settings, feedback, snapshots, kit maps, MIDI library index, renders and the audio-to-MIDI environment stay in `%LOCALAPPDATA%\TakeTown Audio\Hub`, and a later install picks them up. The licence key file (`%APPDATA%\TakeTown Audio\licence.key`) is not removed either.

Things you remove yourself, if you want:

- the loopMIDI ports (or loopMIDI itself, from Installed apps);
- the TakeTown Link and Mackie Control entries in Studio One's External Devices;

- **TakeTown Audio to MIDI**, if you installed it: it is its own entry in Installed apps. Uninstalling it removes the Python environment, the piano model and its own uv, and nothing else (uninstall it before or after TakeTown, either works);
- uv's download cache, if you set Audio to MIDI up by hand: `uv cache clean`;
- the extension in Studio One's program `Scripts` folder, if you answered No above: from an administrator PowerShell, the s1-extension script with `-Target Scripts - Uninstall` (before uninstalling TakeTown, while the script is still there).

14. Appendix

14.1 File locations

What	Path
Plug-ins	<code><plug-in folder>\<name>.vst3</code> ; the plug-in folder is the one chosen in 4.3, by default <code>%LOCALAPPDATA%\Programs\Common\VST3</code> , or <code>C:\Program Files\Common Files\VST3</code>
Where the plug-in folder is recorded	<code>HKCU\Software\TakeTown Audio\TakeTown</code> , value <code>Vst3Dir</code> , and <code>locations.json</code> in the hub folder
Hub folder	<code>%LOCALAPPDATA%\TakeTown Audio\Hub</code>
Hub programs	<code>TakeTownHub.exe</code> , <code>taketown-mcp.exe</code> , <code>vocaltune.exe</code> , <code>vst3params.exe</code> in the hub folder
Install manifest	<code>components.json</code> in the hub folder
Hub log	<code>hub.log</code> , <code>hub.log.1</code> in the hub folder
Hub settings and notes	<code>settings.json</code> , <code>state.json</code> in the hub folder
Integration scripts	<code>integrations\s1-surface</code> , <code>integrations\s1-extension</code> , <code>integrations\reaper-bridge</code> , <code>integrations\claude</code> in the hub folder
Audio to MIDI	<code>audio2midi\</code> in the hub folder (<code>setup.ps1</code> , <code>.venv</code> , <code>models</code>); the download's uv, Python and log in <code>%LOCALAPPDATA%\TakeTown Audio\AudioToMidi</code>

What	Path
Vocal smoothing	<code>vocalsmooth\</code> in the hub folder
MIDI library	<code>midi-library\index.json</code> , <code>midi-library\packs.json</code> in the hub folder
TakeTown Link device	<code>%APPDATA%\PreSonus\Studio One 6\User Devices\TakeTown\Link\</code> (Studio One 7: <code>...\Studio One 7\...</code> ; Studio Pro 8: <code>%APPDATA%\Fender\Studio Pro 8\...</code>)
Compose Tasks extension	<code>%APPDATA%\PreSonus\Studio One 6\Extensions\TakeTownComposeTasks\</code> (likewise)
Extension (admin option)	<code>C:\Program Files\PreSonus\Studio One 6\Scripts\TakeTownComposeTasks.package</code> (likewise under Studio One 7, <code>C:\Program Files\Fender\Studio Pro 8</code>)
Studio One's log	<code>%APPDATA%\PreSonus\Studio One 6\Studio One.log</code>
REAPER bridge	<code>%APPDATA%\REAPER\Scripts\TakeTown\taketown_bridge.lua</code> , registered in <code>%APPDATA%\REAPER\Scripts__startup.lua</code>
REAPER mailbox	<code>%LOCALAPPDATA%\TakeTown Audio\Hub\reaper-bridge\</code>
Claude Code MCP config	<code>%USERPROFILE%\ .claude.json</code>
Claude Desktop MCP config	<code>%APPDATA%\Claude\claude_desktop_config.json</code>
Agents	<code>%USERPROFILE%\ .claude\agents\taketown-mixer.md</code> , <code>taketown-advisor.md</code> , <code>taketown-producer.md</code>
Skills	<code>%USERPROFILE%\ .claude\skills\taketown-mix</code> , <code>taketown-compose</code>
Licence keys	<code>%APPDATA%\TakeTown Audio\licence.key</code>
Song renders	<code><song folder>\TakeTown Renders\</code> (MIDI in <code>TakeTown Renders\MIDI</code>)

14.2 Port names and device settings

loopMIDI port	Studio One device	Field
TakeTown Link to S1	TakeTown > Link	Receive From
TakeTown Link from S1	TakeTown > Link	Send To
TakeTown MCU to S1	Mackie > Control	Receive From
TakeTown MCU from S1	Mackie > Control	Send To

The hub's defaults; they can be changed in the hub's `settings.json` (`"s1link"` and `"transport"` blocks) or on its command line, but the guide assumes the defaults.

14.3 Installer components

id	Name on the components page	Default	Needs
plugins	TakeTown plug-ins (VST3)	always (fixed)	
hub	TakeTown Hub (AI Link)	on	
studio-one-integration	Studio One integration	when Studio One is found	hub
reaper-bridge	REAPER bridge	when REAPER is found	hub
mackie-ports	Virtual MIDI ports (loopMIDI)	when loopMIDI is found (installs nothing)	Studio One integration
claude-integration	Claude integration	when Claude Code or Claude Desktop is found	hub
midi-library	MIDI library index	on (installs nothing)	hub
task-extension-scripts-folder	Compose Tasks extension in Studio One's program Scripts folder	off (admin)	Studio One integration

id	Name on the components page	Default	Needs
audio2midi-pack	Not on the page: the separate download TakeTown-AudioToMidi-Setup		hub

Inside the installer the ids are written with `_` for `-`, which Inno Setup requires (`studio_one_integration`). `install_status` and `install-dev.ps1` use the ids above.

14.4 The integration scripts by hand

Each runs from PowerShell, needs no admin (except `-Target Scripts`), and takes `-WhatIf` to show what it would do and `-Uninstall` to remove what it installed. The two Studio One scripts work on every Studio One found; `-List` shows the folders and `-Version 7` (or "Studio Pro 8") picks some:

```
$hub = "$env:LOCALAPPDATA\TakeTown Audio\Hub"
powershell -ExecutionPolicy Bypass -File "$hub\integrations\s1-
surface\install.ps1"          # TakeTown Link device
powershell -ExecutionPolicy Bypass -File "$hub\integrations\s1-
extension\install.ps1"       # Compose Tasks (per user)
powershell -ExecutionPolicy Bypass -File "$hub\integrations\reaper-
bridge\install.ps1"         # REAPER bridge
powershell -ExecutionPolicy Bypass -File
"$hub\integrations\claudio\install.ps1"          # MCP, agents, skills
powershell -ExecutionPolicy Bypass -File "$hub\audio2midi\setup.ps1"
# Audio to MIDI by hand (needs uv; -Uv <path> for another uv)
powershell -ExecutionPolicy Bypass -File "$hub\vocalsmooth\setup.ps1"
# optional own env for vocal smoothing
```

14.5 Developer notes

Not needed by users.

- **Building from source:** docs/building.md (CMake, MSVC, JUCE; the `release` preset is the one that ships).
- **Building the installers:** docs/building.md, "The installer". They need **Inno Setup 7** (`ISCC.exe` , jrsoftware.org; 6.3 also works), then `installer\build-installer.ps1` writes `build-release\installer\TakeTown-Setup-<version>.exe` and `TakeTown-`

`AudioToMidi-Setup-<version>.exe` (it downloads the pinned uv and checks its hashes). `-StubDir <folder>` compiles both against stand-in files without a full build. The installers, `install-dev.ps1` and `install_status` all read one generated manifest, `components.json`, made by `cmake/Components.cmake`.

- **Installing a local build:** `scripts\install-dev.ps1` installs from a build tree, component by component, the way the installer does (`docs/installing.md`, "Installing from a build").
- **Design docs in the repo:** `docs/installing.md` (what the installer offers, what `install_status` reports), `docs/ai-link.md`, `docs/reaper-bridge.md`, `docs/compose-midi.md`, `docs/sound-design.md`, `docs/licensing.md`.