ABLEREAL V1.0

Mapper Manual

Drive lights, materials and effects from your song

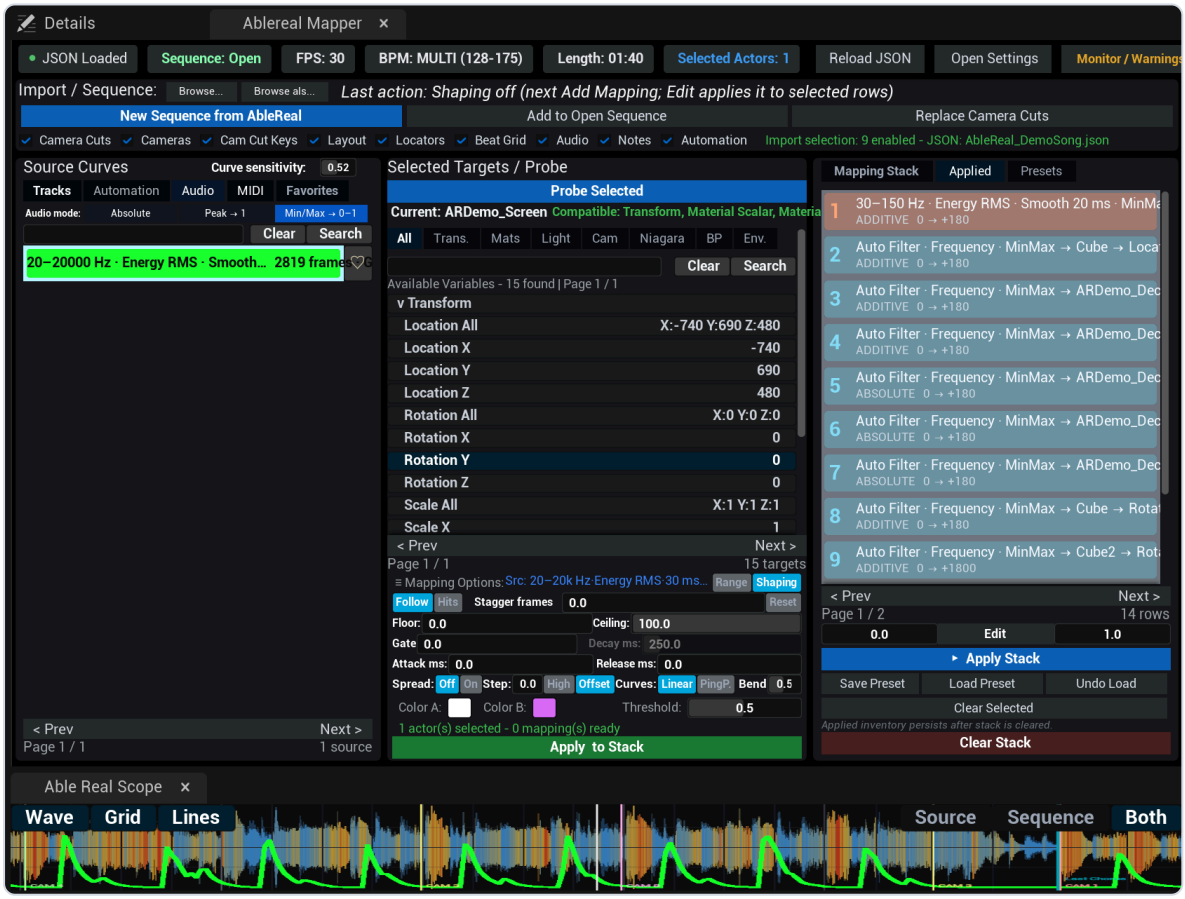
CONTENTS

The four panels
The top bar and Import / Sequence
Source Curves (left)
Selected Targets / Probe (middle)
Mapping Options
The Mapping Stack (right)
The Scope — see a curve before you map it
Driving what the probe can't see
Three recipes
Related

The Director points cameras. **The Mapper drives everything else.**

Any curve in your Ableton set — an automation lane, a MIDI part, an audio level — can drive almost any property of almost any actor in your level. A shelf that slides on the bassline. Lights that breathe with the pad. Bloom that blows out on every snare.

Open it from the menu bar: **AbleReal Shots → Open AbleReal Panels**.



The Mapper: top bar, Import / Sequence, Source Curves, Selected Targets / Probe, Mapping Stack

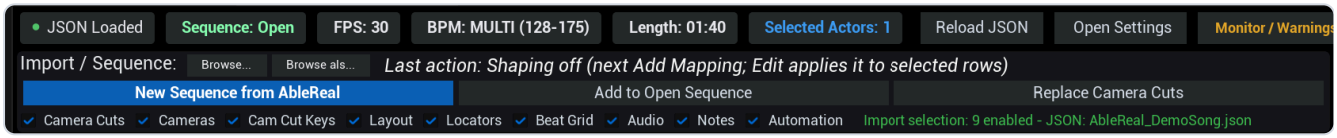
The four panels

The window reads left to right, and so does the job:

```
IMPORT / SEQUENCE       load the JSON, build the sequence
     ↓
SOURCE CURVES           what's driving          (left)
SELECTED TARGETS        what gets driven        (middle)
MAPPING STACK           the list, then apply    (right)
```

You pick a source, select an actor, probe it, choose a property, set the range, and press **Apply to Stack**. Repeat until the stack holds everything you want, then press **Apply Stack** once.

The top bar and Import / Sequence



The top bar and the Import / Sequence row

The **top bar** shows where you are at a glance: **JSON Loaded**, **Sequence: Open**, **FPS**, **BPM**, **Length** and **Selected Actors**. On the right:

- **Reload JSON** reads the selected export again, after you've exported again from Ableton under the same name.
- **Open Settings** opens the folder that holds AbleReal's settings (`YourProject/Saved/AbleReal`).
- **Monitor / Warnings** writes AbleReal's collected warnings, and where its files live, to the Output Log. It's the first thing to press when something looks wrong.

Import / Sequence is where a take is made:

- **Browse...** picks the exported `.json` : the file with exactly the name you typed in Ableton. The files ending in `_base` , `_track_001` , `_automation` ... are its parts. Pick one by mistake and AbleReal switches to the main file; a JSON that is no AbleReal export is refused, with the reason in the status line.
- **Browse als...** points AbleReal at the saved Live set (`.als`) when it can't find it on its own. Tempo changes, mixer automation and clip colours come from there.
- **New Sequence from AbleReal** makes a new take, named after your export (`LS_AbleReal_MySong_Take_01`) and saved in `Content/AbletonImports` .

Which Live set is in use. After every load the status line ends in **Set: MySong.als**, or in **no Live set found (Browse als...)**. AbleReal looks for the set beside the export and in the folders above it, then by the export's name in your Ableton sets folders. A set is only used when it holds at least half of the export's track names, so another song's set is never picked up by accident. With no set, the tempo is taken as constant and the mixer lanes are missing.

AbleReal only reads your export folder. What it reads from the Live set is kept in `YourProject/Saved/AbleReal/AlsCache` , and it is read again by itself when you save the set, export again or change the Curve sensitivity. The export can sit on a read-only or shared disk.

An import that can't start says so. If the selected export is gone or incomplete, New Sequence stops with a message and changes nothing.

- **Add to Open Sequence** writes the import into the sequence that's open.
- **Replace Camera Cuts** rebuilds only the camera cuts of the open sequence.

The checkboxes underneath (**Camera Cuts · Cameras · Cam Cut Keys · Layout · Locators · Beat Grid · Audio · Notes · Automation**) choose what an import writes. Leave them all on for your first run.

Source Curves (left)

Everything your Ableton export brought with it, in five tabs:

Tab	What's in it
Tracks	every exported track, with its curve counts
Automation	automation lanes on their own
Audio	levels captured by `AbleReal_AudioTap`
MIDI	note data
Favorites	the ones you keep coming back to

Each row reads like `Strum — 13 auto · 747 MIDI · 1 audio`, so you can see at a glance what a track actually has to offer. Click one to make it the active source; the Mapping Options strip shows it as `Src: …`.

When a track is selected, the **Automation / Audio / MIDI** tab names light up white for the kinds that track actually has; the ones it lacks stay grey.

Swapping sources is one click. On the Tracks tab, clicking another track keeps the *kind* of curve you had picked:

- an audio range moves to the new track's audio range (the same band, if it has one);
- an automation lane moves to the same device and parameter on the new track (`Auto Filter · Frequency` on KICK → on BASS), or to its first lane if it has no match.

Your Raw / Min/Max choice comes along. MIDI picks stay where they are (a note on one track means nothing on another), and a track that lacks that kind keeps your current pick and says so in the status line. Nothing is mapped until you add it to the stack.



Source Curves: the five tabs and an active source

On the **Audio** tab, **Audio mode** decides how captured levels are read:

Mode	What it does
Raw	the exported 0–1 values as they are
Peak → 1	the loudest captured moment becomes 1
Min/Max → 0–1	the quietest moment becomes 0 and the loudest 1

Quiet tracks barely move anything on Raw; **Min/Max → 0–1** makes them use the full range.

On the **Automation** tab the same strip decides how automation lanes are read, and it starts on **Min/Max → 0–1**. A lane arrives as 0–1 across its parameter's *whole* range, so a filter sweep from 170 Hz to 1.4 kHz would otherwise only travel about a third of it. With Min/Max, the lane's lowest point becomes 0 and its highest 1, so the sweep uses your whole output range, and colour mappings reach Color B. The source label ends in `· MinMax`. Choose **Raw** when a lane's absolute position matters (a lane that should only ever reach half-way). Mappings already in your stack keep the mode they were made with, and presets remember it.

Automation comes straight from your saved Live set: device parameters, rack macros, and the mixer's volume, pan and sends (Master and return tracks included). **Save the set before you export**, and check that the status line names it (see *Which Live set is in use* above).

> **Songs with tempo changes.** Every source follows your tempo changes: MIDI notes, automation lanes and Audio Taps land where the music is, like the camera cuts and the beat grid. A take made with an earlier version keeps its old keys until you press **Apply Stack** again.

Audio Tap captures from an older device. Captures made with the v1.0 device run a little early (about 85 ms). If you keep such an export, open the settings folder (**Open Settings**), set `"audio_offset_ms": 85` in `AbleReal_Settings.json`, then **Reload JSON** and **Apply Stack**. Captures made with the current device are on time and are not shifted.

Curve sensitivity

The number at the top right (default `0.5`) trades key count against accuracy when fitting **curved** automation.

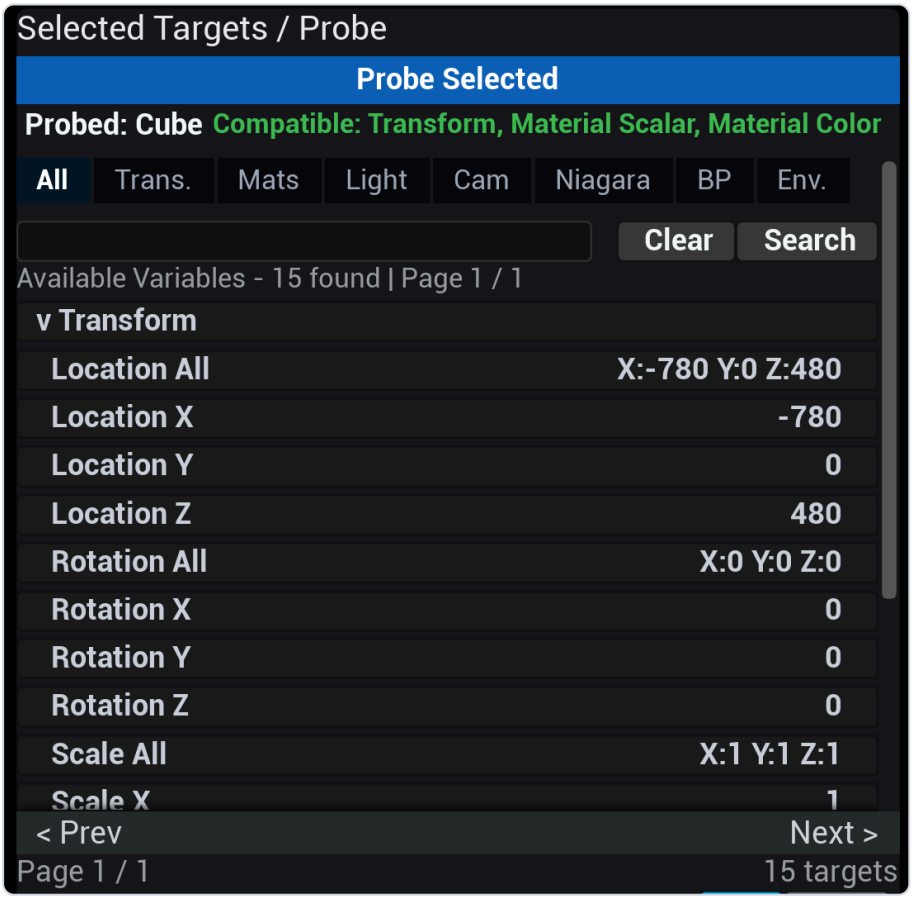
Ableton's bezier segments have to be approximated with keyframes. Low sensitivity gives you fewer keys and a looser fit; high gives you more keys and a closer one. **Hand-drawn linear breakpoints are always exact and are never affected** — this only touches curved segments.

Leave it at `0.5` unless a long, slow curve looks stepped (raise it) or a dense lane is making your sequence heavy (lower it).

Selected Targets / Probe (middle)

Select one or more actors in the level, then press **Probe Selected**.

AbleReal inspects them and reports what it can drive — the header shows the probed actor and its compatible categories, e.g. `Probed: Floor Compatible: Transform, Material Scalar, Material Color`.



Selected Targets / Probe after probing an actor

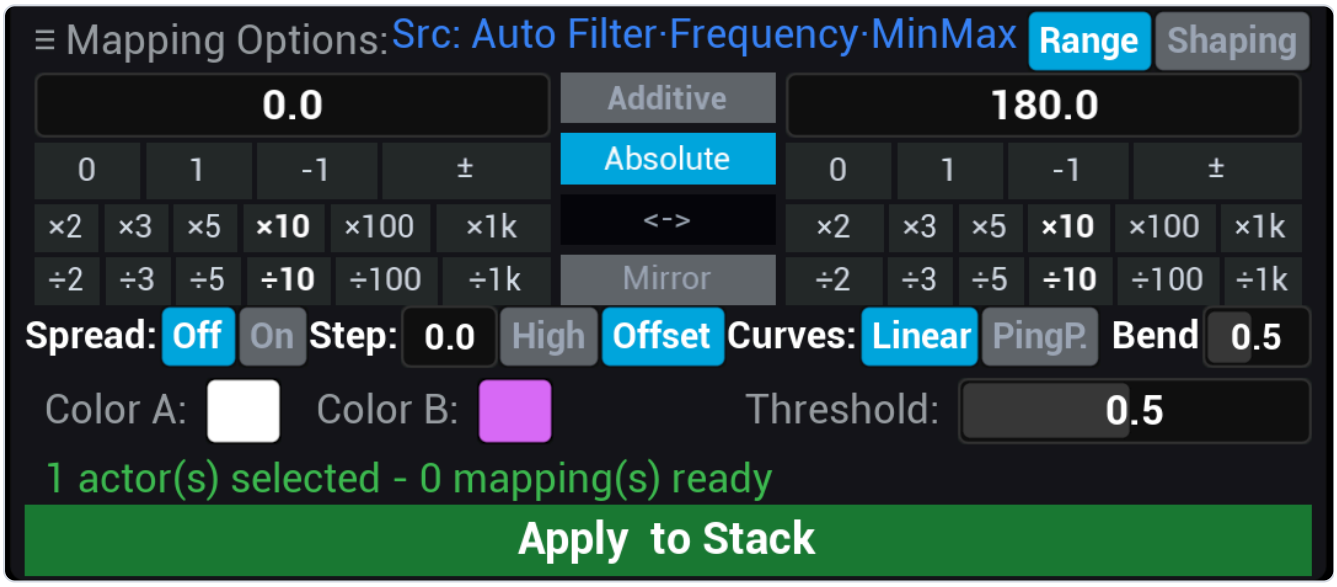
The tabs filter the results: **All · Trans. · Mats · Light · Cam · Niagara · BP · Env.**

Every actor gets Transform for free — `Location All / X / Y / Z`, `Rotation All / X / Y / Z`, `Scale All / X / Y / Z / Uniform`. Beyond that you get whatever the actor really exposes: material scalars and colours, light intensity and colour, camera settings, Blueprint variables.

> The probe deliberately hides internal engine properties and shows artist-useful ones. If something you expect is missing, read the section on adapter actors below — some engine values genuinely can't be reached directly.

Mapping Options

This is the strip under the variable list, and it's where the character of a mapping is decided. It has two tabs: **Range** (where the property goes) and **Shaping** (how the source is cleaned up first). The tab reads `Shaping •` whenever a mapping's shaping is doing something.



Mapping Options: output range, Additive / Absolute, Spread, Curves, Colors, Threshold

The output range

The two number fields flanking the mode button are the **output range** — the values your property will move between as the source goes from its low to its high.

The button grids underneath are fast setters for each side: `0` `1` `-1` `±` for the common values, `×2 ×3 ×5 ×10 ×100 ×1k` and `÷2 ÷3 ÷5 ÷10 ÷100 ÷1k` for scaling what's already there. Between them, `<->` swaps the two ends and `Mirror` mirrors them.

Set a range once, then use `×10` rather than typing — it's much faster to find the right magnitude by ear.

Additive vs Absolute

Absolute means the mapping *owns* the property. The value it writes is the value the property has.

Additive means the mapping *offsets* it. For transforms this is a real structural difference: AbleReal authors a **separate additive Sequencer section** (tinted green) that layers on top of the actor's base transform, rather than one absolute section that bakes in the actor's current position.

Use Additive when you want a thing to stay where you put it and *move around that spot*. Use Absolute when you want the curve to decide where it is.

Additive is the default, and it's usually what you want.

Spread — one curve, many actors

Select several actors, turn **Spread On** and set **Step**. Adding the mapping then staggers the output range across them, so they respond at graduated intensities instead of moving as one block.

- **High** — only the top of the range shifts per actor. The first actor gets the smallest movement, the last the biggest.
- **Offset** — both ends shift together. Every actor moves by the same amount, but around a different centre.

The Step amount is divided evenly across the selection, so the first actor is unchanged and the last gets the full amount.

This is how you get twenty cubes rippling off one bass track instead of twenty cubes doing exactly the same thing.

Stagger (frames) adds *time*: each selected actor reacts a little later than the one before, so a row of lights becomes a chase instead of a block. With Stagger `8` and six actors, they start 0, 8, 16, 24, 32 and 40 frames late. It works with Spread on or off; `Linear` runs first → last selected, `PingP.` runs from both ends to the middle. To change it on mappings you already have, select their rows, set Stagger and press **Edit**; Stagger `0` + Edit removes it.

Threshold

For **bool / checkbox** targets. The source crossing this value flips the property on. A visibility toggle on every kick is `Threshold` plus a drum track.

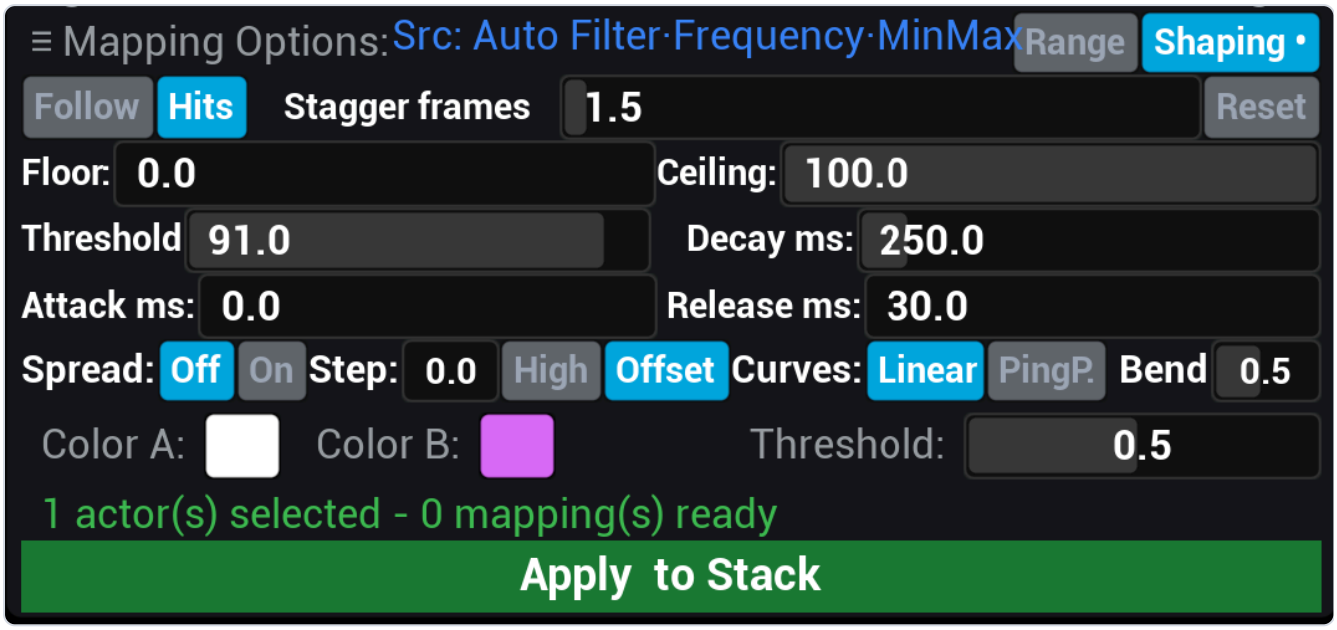
Color A / Color B

For **material colour** targets. The source value blends from Color A at the bottom of its range to Color B at the top, per key.

Curves and Shape

`Linear` or `PingP.` (ping-pong) sets how the source travels through its range, and `Shape` bends it. Use these when a mapping is technically correct but feels wrong — the movement is right, the *timing* of the movement isn't.

Shaping — clean signals, hits and glow



The Shaping tab in Hits mode: Floor, Ceiling, Threshold (Gate in Follow), Decay, Attack, Release, Stagger

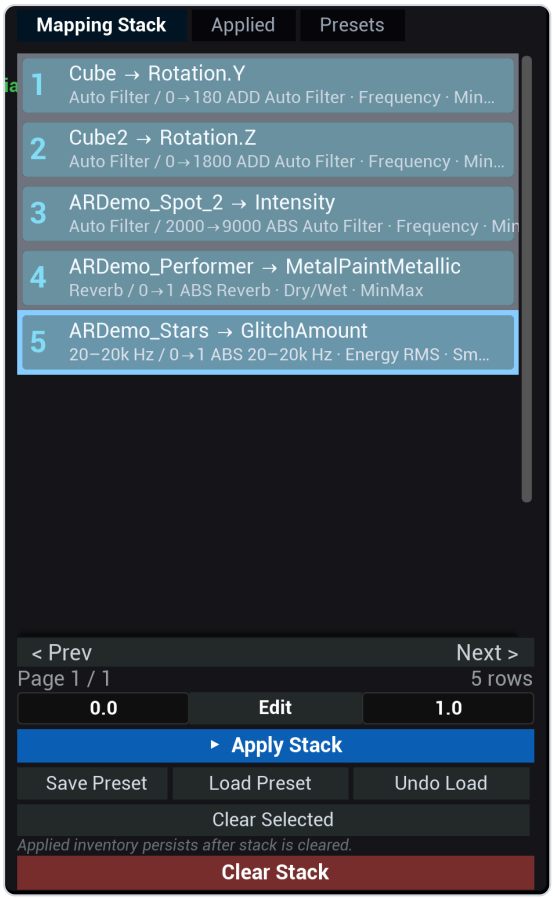
Real audio is messy: reverb tails keep lights half on, busy bands flicker. Shaping cleans the source **before** it reaches your output range, per mapping. When you pick a source, the status line starts with its level, e.g. `Src level 30–95 % (peak 100 %)` — that's where the signal actually sits, so you know where to cut.

- **Floor / Ceiling** — the window you care about. Below Floor the signal is 0, above Ceiling it is 1, and what's between is stretched to the full 0–1. If KICK sits at 30–95 %, a Floor of 30 removes the bleed between kicks and the kicks still swing all the way.
- **Follow** — the curve follows the source (after the window).
- **Gate** (the Threshold box shows *Gate* in Follow): below it the signal is 0, above it passes **unchanged** — no stretch, so it switches on with a step. 0 = off.
- **Hits** — every rise through **Threshold** fires a full flash that falls to 0 over **Decay** (ms). Use it for strobes and punches: one clean hit per kick instead of a wobbling level. Hits fire exactly where the rise begins, because the whole song is known in advance.
- **Attack / Release** (ms) — smooth the rise and the fall. In Follow, Release is the fall time after the gate closes. Release starts at 0 in Follow and 30 ms in Hits; a value you type is kept when you switch.
- **Reset** — back to Follow with no shaping.

The Shaping page is what new mappings get when you add them. Clicking a row in the stack loads its shaping into the page; change it and press **Edit** to re-shape that row (on the Applied tab, Edit re-keys the sequence). The **Scope** shows the result live before you commit to anything.

The Mapping Stack (right)

Apply to Stack adds the current mapping to the list. It doesn't touch your sequence yet — the stack is a shopping list.



The Mapping Stack with a few rows, Apply Stack and the preset buttons

Each row shows the whole mapping in one line:

```
1  SM_WoodenShelf_04 → Location.X
   Robot Glitter / 0 → 1666.665  ADD
```

Select a row and use the two fields plus **Edit** to change its range without rebuilding it. Edit also writes the current Shaping page and Stagger onto the selected rows.

Apply Stack writes every enabled row into the sequence. That's the button that actually changes your film.

Applied actors are filed in the Sequencer under **AbleReal · Mapper**, one folder per source track, tinted with that track's Ableton colour (`KICK` , `BASS` …). An actor driven by two or more tracks moves to a **Multi** folder. Folders you made yourself are never touched.

Presets

Save Preset / **Load Preset** store a whole stack. Build a look you like on one song and reuse it on the next. **Undo Load** backs out a load you didn't mean. Save Preset takes the visible tab: the **Mapping Stack**, or everything on the **Applied** tab.

Loading **replaces** the rows in the stack (nothing reaches the take until **Apply Stack**). **Undo Load** brings back the stack you had before, also after several presets loaded in a row. Each row's source is matched by name to the song you have loaded, so a preset carries over to another song with the same track names.

Your presets are files in `YourProject/Saved/AbleReal/MappingPresets/` ; delete one there to remove it from the list. The **AbleReal Demo Show** preset ships with the demo (`Content/AbleReal/Demo/Presets/`)

and is listed after your own.

Clearing, and what survives

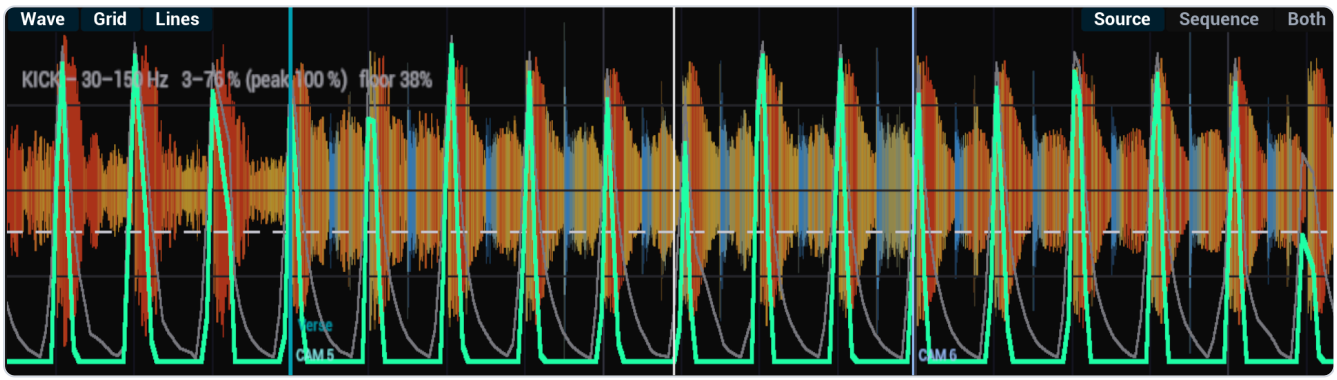
Clear Selected drops the highlighted rows, **Clear Stack** empties it.

The **Applied** tab is a separate inventory of what's already been written into the sequence, and — this is the important part — **it persists after the stack is cleared.** Clearing your working list doesn't undo your work or lose track of it.

Each take has its own Applied list. The list follows the take when you rename it or move it to another folder, and a copy of a take starts with the same list as the original. (Save the take after your first Apply: that is when AbleReal marks it so it can find the list again.)

The Scope — see a curve before you map it

The **AbleReal Scope** panel opens with the others (**Open AbleReal Panels**) and draws curves against your song, scrolling with the Sequencer playhead (the white line in the middle). **4 / 8 / 16** set how many seconds it shows.



The Scope in Source mode: KICK with Shaping, waveform, beat grid, cuts and locators

It has three modes, and it follows what you touched last — click a source in the Mapper and it shows that source; select tracks in the Sequencer and it shows those. The buttons switch by hand.

Mode	What you see
Source	the picked source as it arrives (grey) and after Shaping (in the track's colour), headed `KICK – 30–150 Hz` with its level and shaping. Floor / Ceiling show as light dashed lines, the Threshold or Gate as a red one
Sequence	the curves of the tracks or channels selected in the Sequencer, overlaid, each in the colour of the source that drives it: `Floor · Location.Y ← KICK – 30–150 Hz`
Both	your mapped curves dimmed, plus the source you're auditioning drawn solid on top as `NEW …` — click through sources to see which one fits what's already moving

Behind the curves:

- **Wave** — your song's waveform, Traktor style: bass warm, highs blue. It comes from the song on the take's `AbleReal – Song` track (or an audio file named after your export, next to it). WAV only; the first look takes a moment, after that it loads instantly.

- **Grid** — beats, bars (brighter) and 4-bar phrases (brightest), following your Ableton tempo changes.
- **Lines** — camera cuts in each camera's Director colour, with its name (`CAM 3`), and your Ableton locators in their Sequencer marker colours.

Clicking a camera in the Director never changes what the Scope shows.

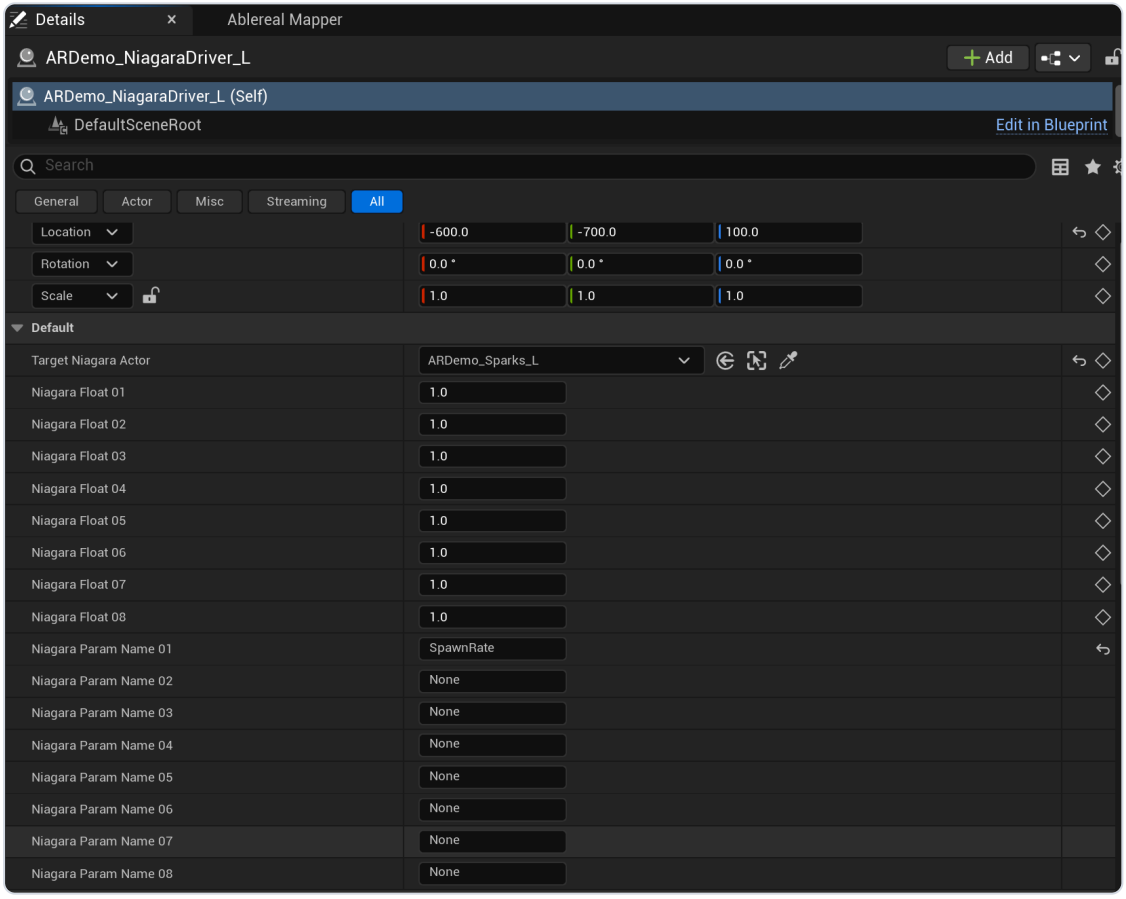
Driving what the probe can't see

Two of the most musically useful things in Unreal are not plain properties, so the probe cannot reach them directly:

- **Niagara user parameters** live in a parameter store, not as normal properties.
- **Post-process grading** lives nested inside a settings struct.

AbleReal ships two **adapter actors** for exactly this. Drop one in your level and it republishes those hidden values as plain floats the Mapper can see. You'll find both in `Content/AbleReal/Blueprints` .

BP_AbleReal_NiagaraSlotDriver — for particle effects



NiagaraSlotDriver in the Details panel: Target Niagara Actor and the parameter names

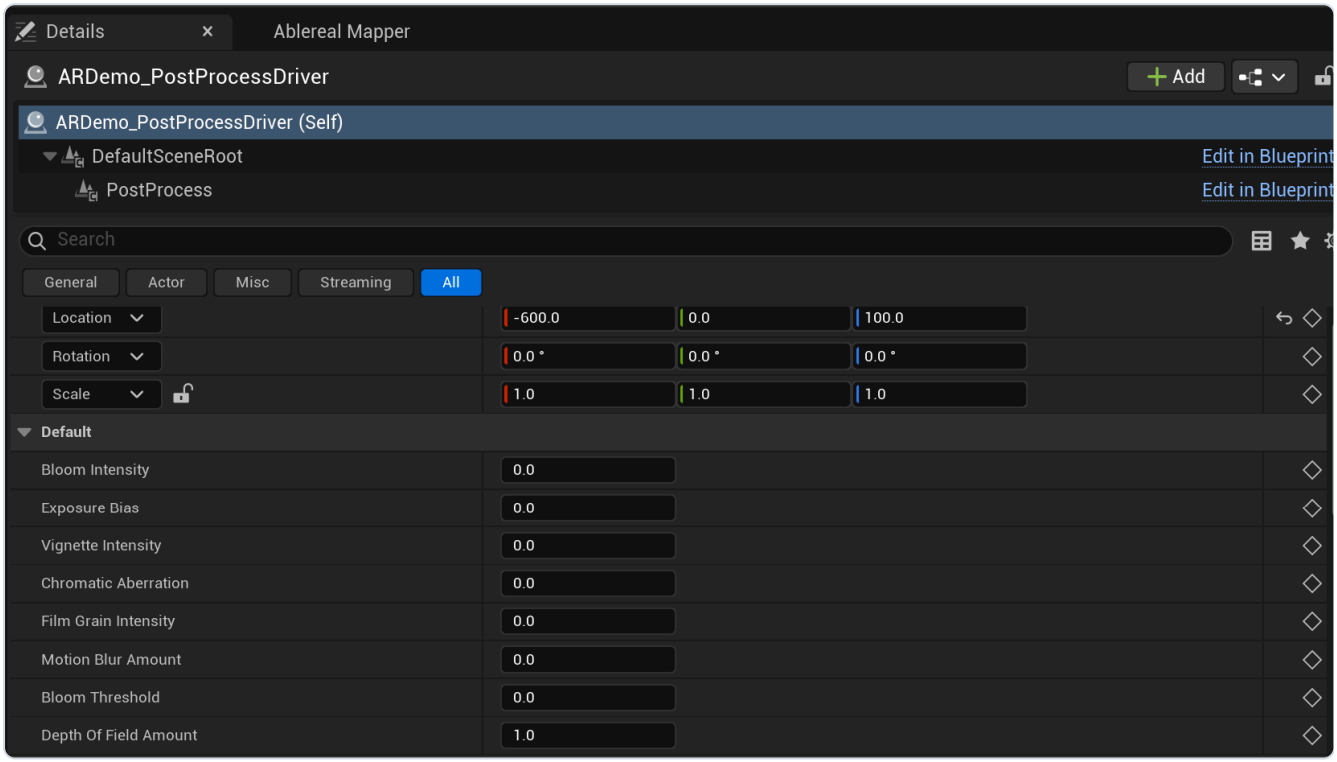
1. Drag it into your level.
2. Set **Target Niagara Actor** to the effect you want to drive.
3. Type the user-parameter names you want into **NiagaraParamName01…08** (they must match the names in your Niagara system exactly).

4. Select the driver, **Probe Selected**, and map **NiagaraFloat01…08** — each one feeds the parameter you named beside it.

Eight slots, any parameters you like. If you select a Niagara actor directly, the probe will remind you to use this instead.

BP_AbleReal_PostProcessDriver — for the grade

Drag it in and it works immediately — it carries its own Post Process Component, so it grades the scene as soon as it's in the level.



PostProcessDriver in the Details panel: its eight grading values

Probe it and you get:

`BloomIntensity` · `BloomThreshold` · `ExposureBias` · `VignetteIntensity` · `ChromaticAberration` · `FilmGrainIntensity` · `MotionBlurAmount` · `DepthOfFieldAmount`

Bloom on the snare, exposure breathing with the pad, chromatic aberration that only appears in the drop. This is the cheapest way to make a scene feel authored rather than rendered — no extra geometry, no extra lights, and it affects the whole frame so it reads on any camera.

Three recipes

The bass shelf. Select the object, probe, `Location.Z`, Additive, output range `0 → 25`, source = your bass track. Small numbers. Almost everything looks better subtler than you first think.

The crowd. Select twenty objects at once. One mapping, `Scale Uniform`, Spread on **High**. One press, twenty different intensities.

The drop. PostProcessDriver, `BloomIntensity` mapped to your lead, output range `0.5 → 4`. Then `VignetteIntensity` on the same source, `0 → 0.6`. The frame tightens and blooms together.

Related

- **Quickstart** — installing, and your first take
- **Director Manual** — the camera half of the product
- **Locator Glossary** — how section names become camera moods