ABLEREAL V1.0

Quickstart

Install, prepare, export, import, direct: your first take in five minutes

CONTENTS

Before you start
1 · Install
Your first take: rebuild the demo (5 minutes)
2 · Prepare your scene
3 · Ableton: capture and export
4 · Unreal: import
5 · Direct it
6 · Render it
Do I need BlackEye?
If something doesn't work
Where to go next

Your song already knows where the camera should be. AbleReal reads your Ableton arrangement and turns it into a cut, moving camera performance in Unreal Engine.

This is the five-minute path. Follow it once end to end and you'll have a finished take.

Before you start

You need	Notes
Windows 10 or 11	The Unreal side is tested on Windows only. Mac is not supported yet
Unreal Engine **5.6 or later**	Earlier versions can't open the assets
Ableton Live 11+ with **Max for Live**	Suite, or M4L as an add-on
A character or object to film	Anything with a mesh works
BlackEye (optional)	See *Do I need BlackEye?* at the end

1 · Install

Ableton

Copy the Max for Live devices into your Ableton User Library. They're in the `AbleReal Max4Live` folder: `1 - AbleReal Max4Live/` in the download zip, or the `AbleReal_Max4Live` zip that comes with the Fab listing.

```
Places → User Library → Presets → Audio Effects → Max Audio Effect
```

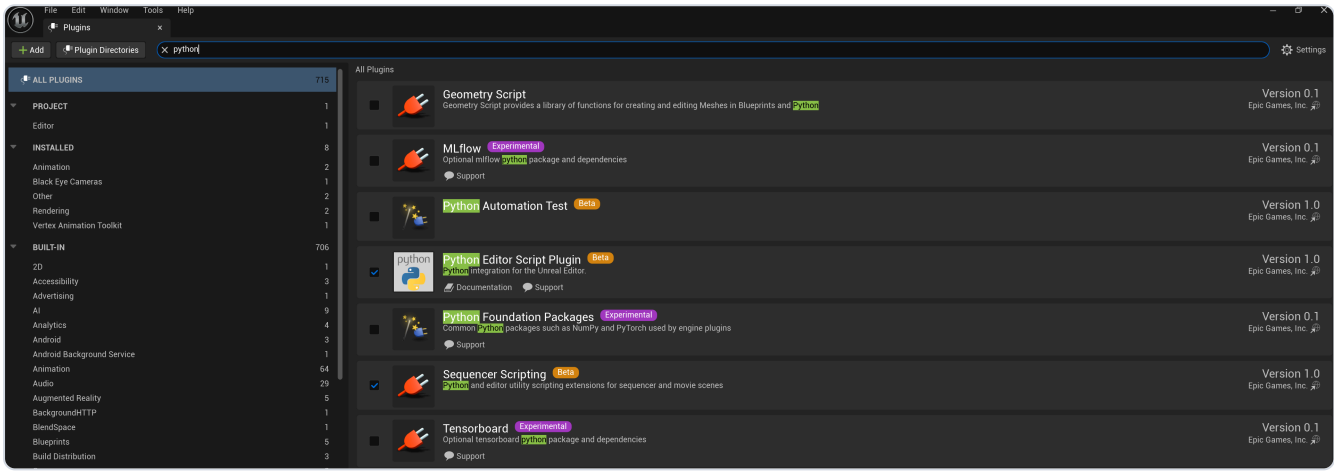
The two devices are **frozen**: their scripts are built in, so each `.amxd` works on its own. Copy the `.js` files along anyway — keep the whole folder together.

The same folder holds `Demo Song`: the demo's Ableton set, its AbleReal export and the song. You'll use it in a minute.

Unreal

First, check that Python is on. It is by default in UE 5.6, but check:

```
Edit → Plugins → search "Python"
  ☑ Python Editor Script Plugin
  ☑ Editor Scripting Utilities
```



Edit → Plugins, searched for "Python", both plugins ticked

Then add AbleReal to your project:

- **From Fab:** in your Fab library (or the Epic Games Launcher), choose **Add to Project** and pick your project.
- **From the download zip:** close the editor and copy the `Content` folder from `2 – AbleReal Unreal/` into your project folder (the one holding your `.uproject`), merging with the Content folder already there.

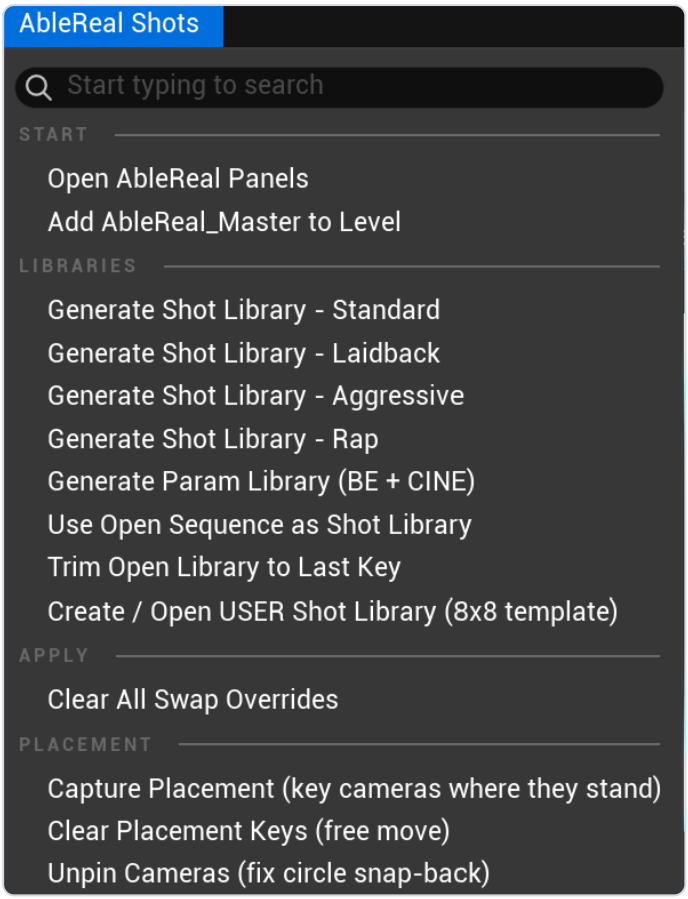
Either way, you end up with:

```
YourProject/Content/Python/
YourProject/Content/AbleReal/
```

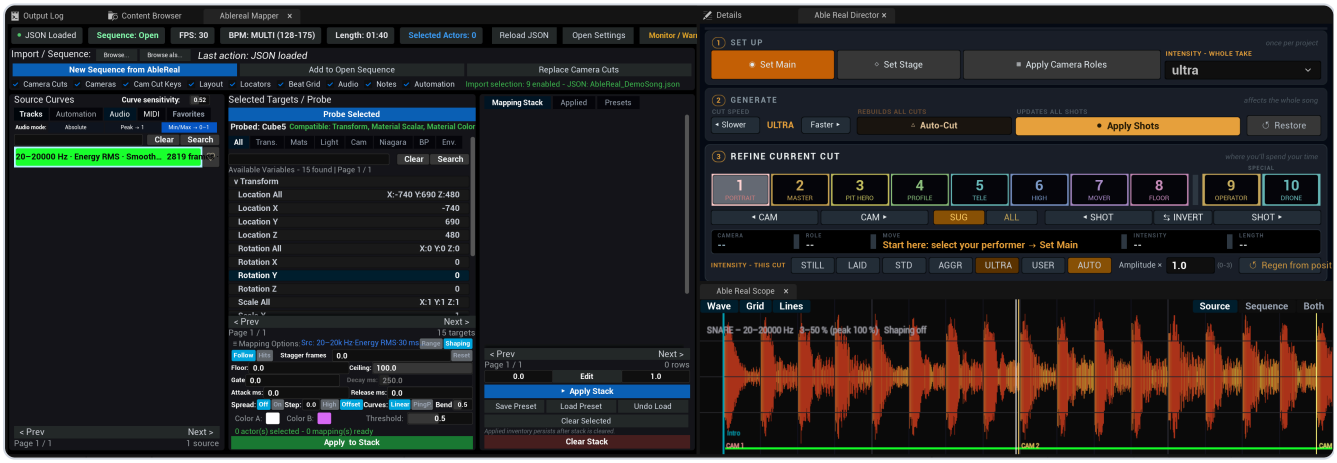
Then open the project again. AbleReal starts with the editor, so it has to be there when the editor opens. A missed restart is the single most common reason AbleReal appears to do nothing.

> **Want to see it working first?** Create a new project from the AbleReal listing on Fab (or open `2 – AbleReal Unreal/Sample Project/` from the zip). The demo level opens with a finished take, ready to play.

Open the panels: in the menu bar, **AbleReal Shots** → **Open AbleReal Panels**. That opens the main AbleReal panel (the **Mapper**), the **Director**, and the **Scope** (a live view of your curves against the song). Dock them wherever you like; Unreal remembers the layout next time.



The AbleReal Shots menu. Start → Open AbleReal Panels



The Mapper and the Director docked side by side

Your first take: rebuild the demo (5 minutes)

The demo level ships with a finished take. Rebuilding it yourself is the fastest way to learn the whole workflow — the stage, performer and lights are already set up, so you only do the AbleReal part.

1. Open the demo level (**AbleReal/Demo/Maps/AbleReal_Demo**), then **File → Save Current Level As...** and save a copy in a folder of your own (for example `Content/MyTakes/AbleReal_Demo_MyTake`). Work in the copy: all takes in a level share its cameras, so this keeps the original demo take exactly as it ships.

2. **Mapper → Browse...** and pick `Demo Song/AbleReal_DemoSong.json` (in the AbleReal Max4Live folder): the one file with exactly that name, not the ones ending in `_base`, `_track_001` and so on. Keep the whole `Demo Song` folder together: the `.als` next to the JSON gives AbleReal the tempo changes (the Peak section is at 175 bpm) and the `.wav` becomes the Song track. The Mapper's status line should now read **JSON loaded - Set: AbleReal_DemoSong.als**.
3. **New Sequence from AbleReal.** You get a fresh take with cameras, camera cuts, beat grid and the song.
4. **Director:** select the performer → **Set Main**; select the video wall → **Set Stage** (the two wide cameras now frame the show).
5. **Auto-Cut.** It cuts the song and gives every cut its camera move. Try the Intensity menu and a few chips — these are your camera decisions.
6. **Mapper → Presets → AbleReal Demo Show → Load**, then **Apply Stack.** The whole light show — screen, LED poles, arches, visualizer, sky — lands on the take.
7. Press play.

Curious how the song drives all that? Open `Demo Song/AbleReal_DemoSong.als` in Ableton: the **VIS** track's macros (Sun, Hue, Sweep…) are drawn by hand, and an **AbleReal_AudioTap** sits on each instrument track.

---

2 · Prepare your scene

Three things, and they're the ones that trip everyone up.

Your performer's root must sit at the floor

AbleReal builds every camera move relative to your subject's origin. If the root is at the hips or floating above the ground, framing will be off by that amount for the whole song.

Select your character and check that the pivot is at its feet.

Name your cameras so AbleReal can find them

The rule is forgiving: **any camera whose name contains "cam" and ends in a number.**

✅ `Cam1` · `Camera 3` · `CineCameraActor2` · `Black_Eye_LookAt_CineCam5`

❌ `HeroCam` (no number) · `Cam1_wide` (doesn't end in the number)

Any actor that *has* a camera component counts, so Blueprint camera rigs work too. Cameras 1–8 are the main set; 9 and 10 are the special ones.

Don't have cameras yet? Skip this. AbleReal creates all ten for you: cameras 1–8 when you import (step 4), and cameras 9 and 10 as soon as it knows what to point at (**Set Main**, just below).

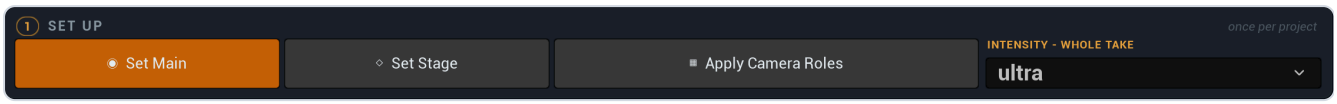
Tell AbleReal what to point at

In the **Director** panel, section **1 · SET UP**:

1. Select your performer in the level → click **Set Main**
2. Click **Apply Camera Roles**. This positions the cameras into their roles (hero, wide, profile, and so on) around your subject, on the side your performer faces, and makes sure cameras 9 and 10 exist

Set Stage puts the show in the shots. Select what your show is built around (a video wall, the back of the stage, the lighting rig) and click **Set Stage**: the two wide cameras (2 and 6) frame it instead of the

performer, and the cameras face the stage from the right side. Without it, every camera films the performer.



The Director's SET UP row: Set Main, Set Stage, Apply Camera Roles

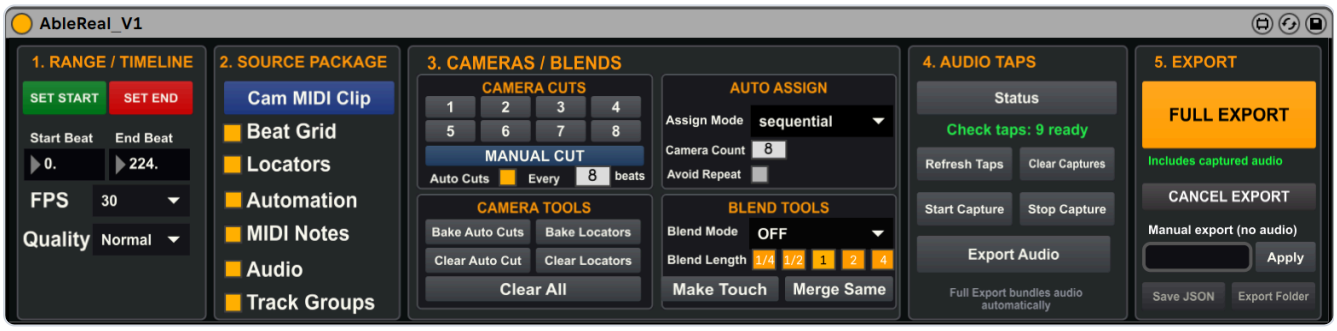
That's your scene ready.

> **Quickest of all: select your performer, then import.** If there's no MAIN target yet and your performer is selected when you click **New Sequence from AbleReal** (step 4), AbleReal sets Main on it first and places all ten cameras around it in one click.
>
> If you import with nothing selected, cameras 1–8 wait at the world origin, the Director's status strip says **Start here** with Set Main lit up, and chips 9 and 10 read **SET MAIN**. Select your performer and press **Set Main**: that one click aims the cameras, puts them in their roles and adds cameras 9 and 10.

---

3 · Ableton: capture and export

Drop `AbleReal_V1` on any track. The device is numbered — work down it in order. Hover any control and Live's **Info View** (bottom-left) explains it in one line.



The AbleReal_V1 device, sections 1 to 5

1 · RANGE / TIMELINE — set the part of your arrangement to export. The range can start anywhere in the song.

2 · SOURCE PACKAGE — choose which tracks and automation come along. These become the "source curves" you can map to anything in Unreal.

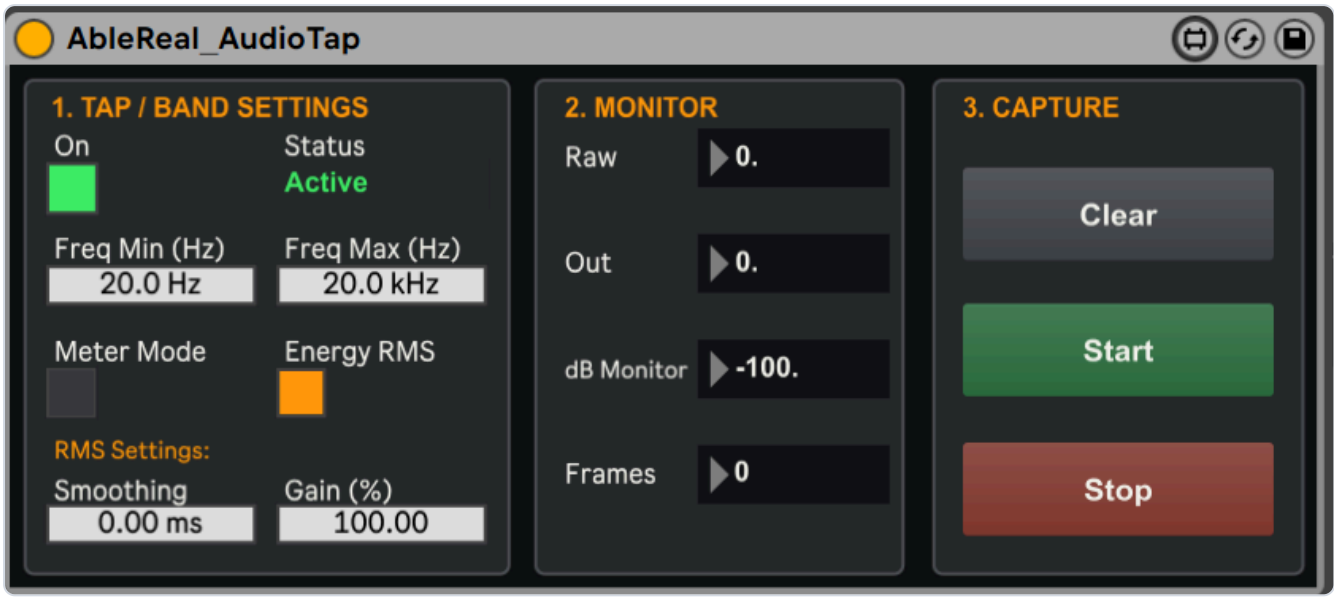
3 · CAMERAS / BLENDS — set your **Camera Count**, then create the camera MIDI clip. Each note in that clip is one camera cut: pitch chooses the camera, length sets how long it holds.

You can also let AbleReal place cuts for you from your **locators** — which is the fastest way to get a good first take. **Bake Auto Cuts** adds a cut every **8 beats** by default (2 bars); try 16 for a calmer edit.

Your cuts live in the **AbleReal_Camera_Cuts** track, not in the device. Delete or replace the device and a new one finds that track (keep its name) and takes its Start and End from the camera clip.

4 · AUDIO TAPS — optional. Add `AbleReal_AudioTap` to individual tracks to capture their audio for use as animation sources. Each tap listens to one frequency band (a kick: about 30–150 Hz). Leave the response on

RMS, the default; the device remembers its settings with the set. **Gain** is a percentage: 100 % leaves the level as it is.



An AbleReal_AudioTap: band, RMS settings, monitor and capture

5 · EXPORT — hit **FULL EXPORT** and choose a name and folder. It bakes automation, scans your tracks, then plays the range once to capture audio. The line under the button shows the progress (*Finding automation*, *Baking automation 42%*, *Scanning tracks*, *Capturing audio 63%*) and then *Export complete*. **CANCEL EXPORT** lights up while an export runs and stops it cleanly. If a button seems to do nothing, read that line: errors such as *Set Start first* appear there in red. (**Manual export (no audio)** writes the JSON to the **Export Folder** without the audio pass.)

> ⚠️ **Save your Live set before exporting, and keep the export with it.** AbleReal reads tempo changes and mixer automation straight from the saved `.als` file. It finds the set when the export sits **in your Live project folder** (next to the `.als`, or in a folder inside it) or **carries the set's name** (`MySong.json` for `MySong.als`). After you load the export, the Mapper's status line reads **Set: MySong.als**. If it reads **no Live set found**, the tempo is taken as constant: point AbleReal at the set with **Browse als...**.

Exporting again. Press **FULL EXPORT** a second time and it uses the same name without asking: the new export replaces the old one. In Unreal, press **Reload JSON**, then import.

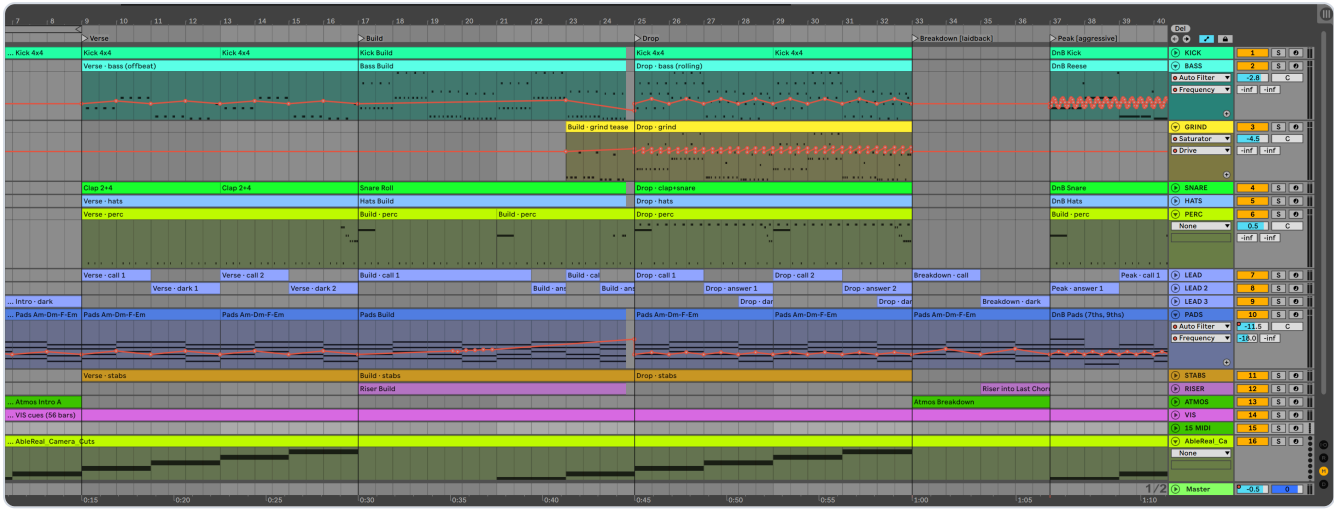
A range that starts later in the song. Live starts playing at bar 1 for a moment and then jumps to your start. That is normal: the Audio Taps record from your start, not before.

Your song in the take. Render the song from Ableton (**File → Export Audio/Video**) as a **WAV** with the same name as your export, saved next to it: `MySong.wav` beside `MySong.json`. Render **exactly the range you exported**, from the same start to the same end: the song is placed at the start of the take. Every new take then plays it on an **AbleReal - Song** track, and the Scope draws its waveform. Render it again under the same name and the next take you make uses the new file.

Bars are four beats. AbleReal counts every bar as four beats. A song in 3/4 or 6/8 still cuts on the beat, but the cuts and the grid do not follow its bar lines.

Name your locators — it matters more than anything else here

AbleReal reads your arrangement locators and gives each section a **mood**, and the mood decides how the cameras behave. Name them the way you already do — `Intro`, `Verse 1`, `Chorus`, `Drop`, `Breakdown` — and it just works.



An arrangement with named locators: Intro, Verse, Chorus, Drop

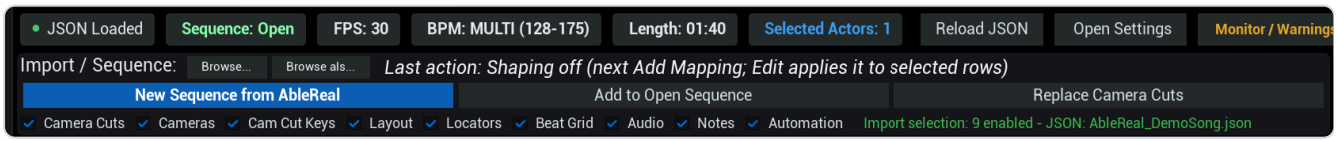
A song with named sections looks dramatically better than one without. See the **Locator Glossary** for the full vocabulary and the `[style]` tags.

---

4 · Unreal: import

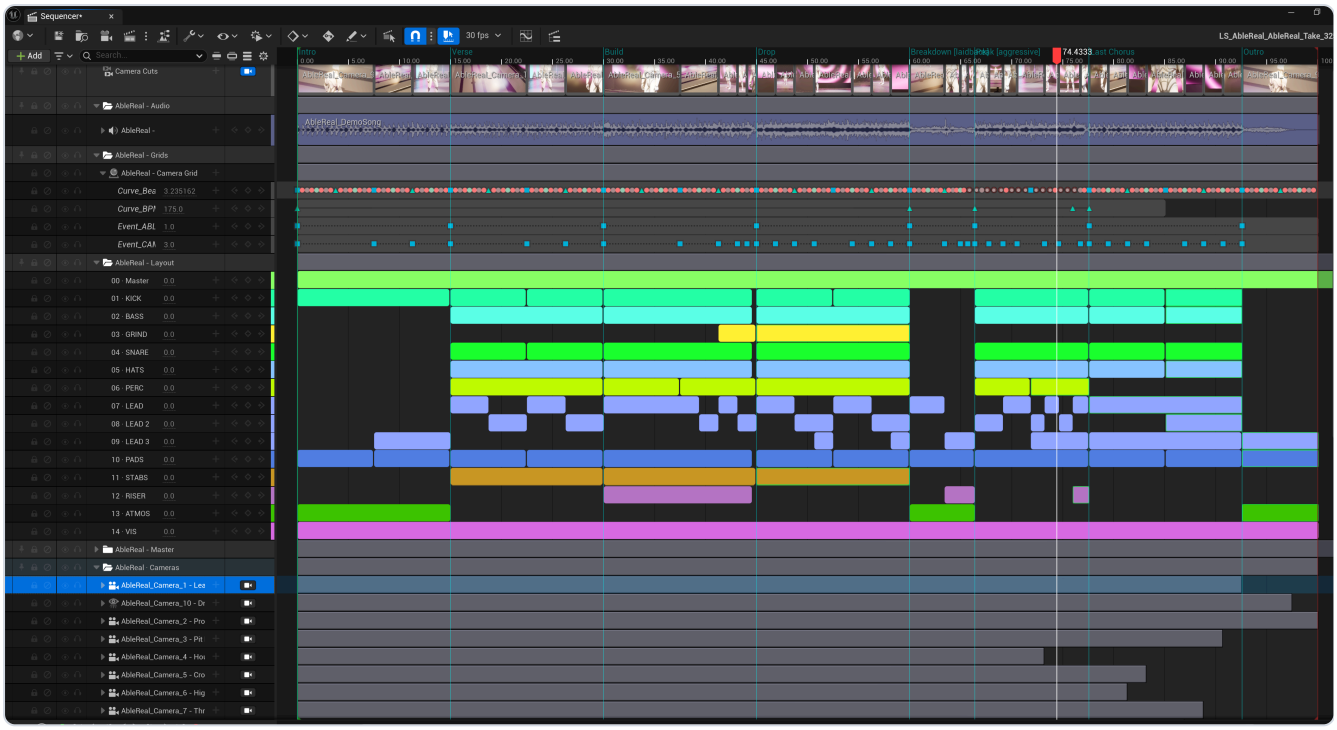
In the **Mapper**:

1. **Browse…** → pick the JSON you just exported. An export is about twenty files; pick the one with exactly the name you typed in Ableton (`MySong.json`). The others (`MySong_base.json`, `MySong_track_001.json` …) are its parts. If you pick a part, AbleReal switches to the main file and says so. The top bar should turn to **JSON Loaded** and show your FPS, BPM and length, and the status line names your Live set.
2. Leave the import checkboxes as they are for your first run.
3. Select your performer, then click **New Sequence from AbleReal**.



The Mapper's top bar and import row: JSON Loaded, New Sequence from AbleReal

You now have a Level Sequence with all ten cameras, your cuts, your beat grid and your audio. Press play.



A new take in Sequencer: cameras, camera cuts, beat grid and song

The beat grid, section markers and mapped curves live on an actor called `AbleReal_Master`. If your level doesn't have one yet, the import places it for you. **Save the level** so it stays.

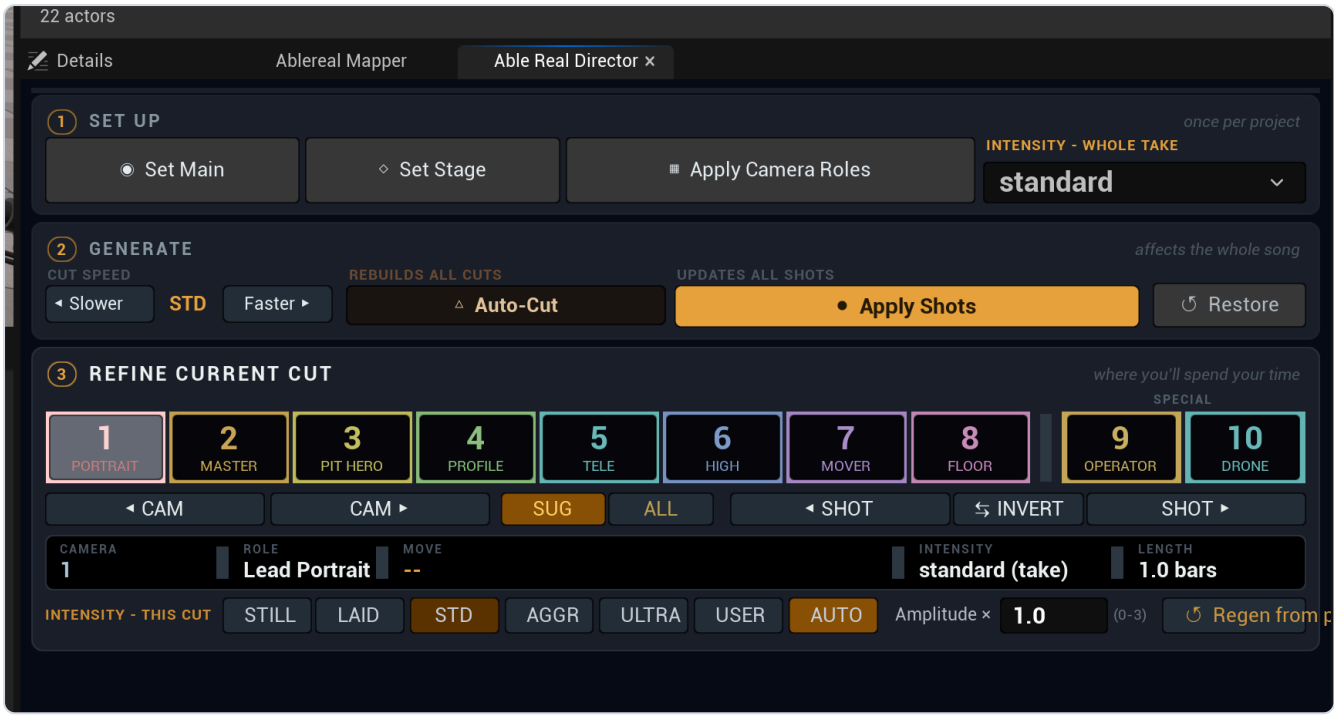
Every import makes a new take, named after your export (`LS_AbleReal_MySong_Take_01`, `_Take_02`…) and saved in `Content/AbletonImports`, so nothing you did before gets overwritten.

The first time in a new level, AbleReal may stop and ask you to place one camera: *place one Cine Camera Actor*. Drag one in from **Place Actors** → **Cinematic** (anywhere), then click New Sequence again. Unreal 5.6 can crash when a script creates the first camera in a level; one you place by hand is safe, and AbleReal copies it from then on. You only do this once per level.

---

5 · Direct it

The Director panel is where it becomes yours. Working down:



The Director panel

2 · GENERATE

- **Auto-Cut** rebuilds the cutting rhythm from your song's structure, with every cut landing exactly on the beat. It also gives cameras 9 and 10 their big moments (see *The money cams* below).
- **Apply Shots** gives every cut a camera move from the shot library. The first time, AbleReal picks its built-in *standard* library for you; the **Intensity – Whole Take** dropdown switches to another one.
- **Cut Speed** (◂ Slower / Faster ▸) makes the whole take cut slower or faster
- **Restore** goes back one step if you don't like what you got: the take as it was before your last Auto-Cut or Cut Speed change. Press it again to go further back

The money cams. Cameras 9 and 10 stay out of the normal rotation. After every Auto-Cut, AbleReal hands them a few moments that deserve them: the **Operator** (9) takes the entry of a chorus or showcase, the **Drone** (10) takes long, calmer stretches. Both only take cuts of 4 bars or more. How many you get follows the whole-take intensity, from one at *laidback* up to five at *ultra*.

3 · REFINE CURRENT CUT — scrub to any cut and reshape just that one:

- The **camera chips (1–10)** switch which camera owns the cut; ◂ **CAM / CAM** ▸ step through them
- ◂ **SHOT / SHOT** ▸ cycle to a different move
- **INVERT** mirrors the move — same motion, opposite side
- **Intensity – This Cut** (STILL · LAID · STD · AGGR · ULTRA · USER · AUTO) changes the vocabulary the move is drawn from
- **Amplitude ×** scales how big the moves are on that camera

Re-run **Apply Shots** as often as you like. It's not destructive, and every run reshuffles within the rules — so if a take doesn't sing, run it again.

---

6 · Render it

Movie Render Queue is switched on in this project. Open **Window** → **Cinematics** → **Movie Render Queue**, press **+ Render** and pick your take, choose where the files go, then press **Render (Local)**.

The take renders with its own camera cuts. The demo level's own sequence does not start by itself, so only the take you render plays.

Do I need BlackEye?

No. Without it you get all ten cameras: eight main cameras plus the two special cameras, `AbleReal_Camera_9 - Operator` and `AbleReal_Camera_10 - Drone`, running on AbleReal's own baked motion.

With BlackEye installed, cameras 9 and 10 (same names) are created as BlackEye rigs instead and get the full rig feature set: live target tracking, focus handoff and attention shifts that follow your performer.

Already have your own camera labelled 9 or 10? AbleReal uses it as it is and doesn't create another.

If something doesn't work

The panel opens but nothing happens when you click. Python isn't enabled, or you didn't restart the editor after enabling it. Go back to step 1.

No cameras appear. Cameras 1–8 are created by **New Sequence from AbleReal**. If you placed your own, check the naming rule in step 2.

The cameras sit around the middle of the world, aimed at nothing. You imported before Set Main. Select your performer and press **Set Main** in the Director; the first time after an import it also applies the camera roles.

Only eight cameras, no 9 and 10. Same cause, same fix: Set Main. Chips 9 and 10 read **SET MAIN** until it's done.

The cameras film the performer from the side or from behind. Select a point at the back of the stage, behind your performer, and click **Set Stage**: the cameras move to that side by themselves.

The moves look off / framing is wrong. Your performer's root probably isn't at the floor.

The import worked but there's no beat grid or section markers. The log says `Could not find any AbleReal bus actor`: AbleReal couldn't place `AbleReal_Master` for you. Use **AbleReal Shots** → **Add AbleReal_Master to Level**, then import again.

Timing drifts on a song with tempo changes. Look at the Mapper's status line. If it reads **no Live set found**, AbleReal could not find your saved `.als`: save the set, keep the export in the Live project folder, or use **Browse als...**. See Troubleshooting for the rest.

See the **Troubleshooting** guide for the longer list.

Where to go next

- **Locator Glossary** — the mood vocabulary, and `[style]` tags for giving one section a different character
- **Director Manual** — every control on the Director panel
- **Mapper Manual** — driving lights, materials and effects from your song

Then do the thing that actually teaches you what AbleReal is: take one song, set **Intensity – Whole Take** to *laidback* and press **Apply Shots**, watch it, then do the same at *ultra* and watch it again. Same song, same cuts, completely different film.

Ableton, Live and other Ableton product names are trademarks of Ableton AG. Unreal and Unreal Engine are trademarks or registered trademarks of Epic Games, Inc. AbleReal is an independent product, not affiliated with or endorsed by Ableton AG or Epic Games.