

SimpleCue user guide

SimpleCue is an audio cue player for theatre, live events and installation. You build a list of cues and press **GO**.

Before a show: the playback engine is verified sample-by-sample and the control layer is driven end-to-end over real sockets — 380 checks in all. But SimpleCue **has never been run on a live show**. No MIDI, lighting or streaming hardware has ever been connected to it, and only the macOS/CoreAudio build has been used with real audio hardware. Rehearse with it before you trust it with a performance.

The basics

A **cue** is an audio file with an in point, an out point, a gain and a pre-wait. The cue list runs top to bottom, and **GO** fires whatever is on standby.

Everything else is about how cues start, stop, and hand over to each other.

The screenshot displays the SimpleCue software interface. At the top, there's a 'GO (space)' button and a '3' indicating the current cue. Below this, a row of buttons includes 'Stop all (S)', 'Pause (P)', 'Release vamp (enter)', and 'PANIC (esc)'. The main area shows a list of cues with columns for Cue, Name, Source, Pre, Length, Fades, Repeat, and Link. Cue 3, 'Storm builds', is highlighted with a yellow 'VAMP 1' badge. Below the cue list, a waveform view shows the audio signal with 'IN' and 'OUT' markers. The 'SOURCE' panel at the bottom shows details for Cue 3, including the name 'Storm builds', the audio file 'storm-builds.wav', and the gain '0.0 dB'.

Cue	Name	Source	Pre	Length	Fades	Repeat	Link
1	Pre-show bed	preshow-bed.wav	00:00.00	open	6.0s in	loop	
2	Interval music	interval-bed.wav	00:00.00	00:44.0	2.0s in / 5.0...		xfade 6.0s -> next
3	Storm builds	storm-builds.wav	00:00.00	open	3.0s in	vamp	at end -> next
	Play cue	3.0s fade in	00:00.00	00:03.00			
	Devamp	00:03.0 to 00:11.0, finishes the pass					
	Fade/Stop	3.0s fade out		00:03.00			
3.5	Door slam	door-slam.wav	00:00.40	00:02.5			
4	Blackout - fly LX and video	2 messages	00:00.00	open			
5	Walk-out playlist	Spotify House walk-out	00:00.00	open	4.0s in		
6	Act 1 preset	preshow-bed.wav	00:00.00	01:36.0	3.0s in		
7	Rain exterior	storm-builds.wav	00:00.00	00:42.0	2.0s in / 4.0...		instantly -> next
7.5	Distant bell	door-slam.wav	00:02.50	00:02.5			
8	Kitchen radio	interval-bed.wav	00:00.00	01:04.0	1.5s in / 2.0...		
9	Radio off	door-slam.wav	00:00.00	00:02.5			
10	Night ambience	preshow-bed.wav	00:00.00	01:36.0	8.0s in		
11	Clock strikes three	door-slam.wav	00:00.00	00:02.5			at end -> next
12	Drum canyons	storm-builds.wav	00:00.00	00:42.0	6.0s in / 6.0...		xfade 6.0s -> next

Cue 3 is vamping — the yellow **VAMP 1** badge on the row, its **Devamp** sub-cue exposed beneath it, and **Release vamp** lit in the transport. Note the cue numbers: **3.5** and **7.5** sit between whole

numbers, the way a prompt book does.

Cue numbers

Cue numbers are **free text**, not just integers: 12 , 12.5 , PRE , INTERVAL . Insert a cue between 12 and 13 and call it 12.5, the way a real prompt book does.

Fades

Each cue has a **fade in** and a **fade out** time, and a **shape** — five curves are available. A linear fade sounds like it dips in the middle; the other shapes exist because different material wants different curves.

Show-wide defaults for fade times and shape mean you set your house style once.

Links: how one cue leads to the next

This is what turns a list into a show.

- **Fire the next cue immediately** — the moment this one starts.
- **Fire at the end** — an auto-follow.
- **Crossfade into it** — this cue fades out while the next fades in, over a set duration.

A link with **no target** means *the next cue in the list*, so inserting a cue between two linked cues does what you'd expect rather than breaking the chain.

Delay is ignored on a crossfade. The crossfade's own duration governs the timing.

Loops

Repeat a section a set number of times — or **forever**, which is what a loop count of zero means. Useful for pre-show music and ambient beds.

Vamps — the one worth understanding

A **vamp** circles a section of a cue until the operator calls for it to continue.

This is the feature that handles live theatre not running to time. Underscore a scene change, or hold music under an entrance, and it goes round and round until you press GO — then it leaves the loop and continues, rather than cutting.

Set the vamp region (start and end) and enable it — the **Vamp from** and **Vamp to** fields under the waveform, drawn on it as the shaded region between the **V<** and **>V** markers. While vamping, the cue's **Devamp** step appears; GO releases it.

Cue lifecycles and GO

A cue expands into the steps it actually needs:

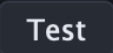
- **Play** — always
- **Devamp** — only if the cue has a vamp
- **Fade / Stop** — always

GO walks these one at a time. That's why a vamping cue takes an extra GO to release: the first GO devamps, the next moves on.

By default, firing a cue also plays it, and standby skips straight past the Play step — so you're never offered the same audio twice.

Output routing

Each cue has a routing matrix mapping **source file channels to device output channels**, with a gain per point. A stereo file can go to outputs 3 and 4, a mono effect to a single speaker, one channel can feed several outputs.

Output: MacBook Pro Speakers  

Input: << none >>  

Active output channels: ☒ 1 + 2

Sample rate: 96000 Hz 

Audio buffer size: 512 samples (5.3 ms) 

☐ Enable inputs (needed for streaming loopback capture)

96.0 kHz, 512 sample buffer (5.3 ms)

2 output channels, 0 inputs

Changing the sample rate reloads every cue's audio at the new rate.

Audio is resampled to the device rate **at load time**, which is what makes the loop and vamp maths exact integer sample arithmetic. The consequence to know about: **changing the device sample rate reloads every cue in the show**. Do it before a show, not during one.

Cues without explicit routing get a sensible default from the file's channel count and your device's outputs.

Remote control

SimpleCue can be driven by, and can drive, other equipment:

- **OSC** — in and out, with Bitfocus Companion support
- **MIDI** — notes, CCs and program changes
- **MIDI Show Control (MSC)** and **MIDI Machine Control (MMC)**
- **Art-Net and sACN** — trigger cues from a lighting desk

Full details in `control.md`.

Two things to know as an operator:

Lighting-desk triggers are edge-detected. A cue fires when the level *crosses* the threshold, not on every frame the desk is sending. Without that, a desk holding a channel at full would re-fire the cue continuously.

DMX addresses cues by their position in the list, not by cue number. DMX can only count, so it has no way to send `12.5`. Every other protocol uses the cue number. **This means inserting or deleting a cue shifts what your DMX triggers point at** — re-check your triggers after editing the list.

Show files

A show saves as a single `.cueshow` file: plain, readable JSON.

- **Audio paths are stored relative to the show file**, so a show folder can be copied to another machine and still work.
- **Saving is atomic** — written to a temporary file and moved into place, so a crash or a pulled USB stick mid-save cannot destroy the show already on disk. At a venue that file is often the only copy.
- Shows saved by an older version open fine. A show saved by a *newer* version is **refused** rather than partially loaded — better an honest error than a show that quietly behaves differently from the one you built.

Older `cue-player` show files still open.

Practical notes

Changing audio device sample rate reloads every cue. Audio is resampled once, at load, which is what makes all the loop and vamp timing exact. Expect a pause, and do it before the show rather than during.

Set your audio device first, then build the show.

Platforms

macOS (universal), Windows (x64 and arm64), Linux (x64 and arm64). Only the macOS/CoreAudio build has been exercised against real audio hardware.

SimpleCue · v0.3.2 · guide updated 2026-08-02 · stoatworks-labs.com/software/simplecue/

Latest version of this guide: stoatworks-labs.com/software/simplecue/guide/ · Source and issues:

<https://github.com/stoatworks-labs/simplecue>