



VERDALIS

Natural sound synthesis

CrackleBlaze

Synthesised fire



Version 0.1.0

CLAP instrument for Linux and Windows

Part of the Verdalis Plugin Suite

2026-09-11

Contents

Introduction

- A fire is not a river
- How a crackle arrives
- What is in this manual

Installing

- Linux
- Windows
- Hosts
- Your own presets
- Building from source

A first sound

- Three things to try first

How it works

- The roar, and where its colour comes from
- A fire flares as one flame
- A crackle is a click, not a ring
- A sizzle is a crackle held open
- A log giving way
- Nothing below 60 Hz
- Distance and the hearth

The plugin window

- Gestures
- The mixer

Parameter reference

- Blaze
- Crackle
- Sizzle
- Settle
- Hearth
- Filter
- Envelope
- Output

The preset library

Using it in a host

- Notes
- Automation and modulation
- Reproducible renders
- CPU cost
- Rendering and tails
- Headroom

What is still being fitted

Reference and limits

- Where the model comes from
- What it does not do

Introduction

CrackleBlaze is a CLAP instrument that generates fire: campfires, hearths, wood stoves and large open blazes. It contains no samples. The combustion roar, every crackle, every hiss of steam out of wet wood and every thump of a log giving way are computed from noise, filters and envelopes while the plugin plays, so no two takes are alike and there is no loop point to find.

A fire is not a river

One number decides the whole design, and it is the first thing the reference library says.

	CRACKLEBLAZE'S LIBRARY	RIVERFLOW'S LIBRARY
crest factor, median	31.7 dB	19.6 dB
envelope variation at 4 ms	0.85	0.26
envelope variation at 50 ms	0.56	0.12

Half of a river is its bed — a third of RiverFlow’s reference library is statistically indistinguishable from shaped noise. A fire is the opposite: a *quiet* bed with very loud, very short things happening on top of it. Twelve decibels of crest factor separate the two, and almost every decision in this plugin follows from that ratio.

So the roar here is deliberately modest, and the layers above it carry the sound. Turning *Crackle Level* down until the fire is only its roar is not a broken patch — it is a fire heard from far enough away that nothing resolves any more, which is what *Distant Fire* is.

How a crackle arrives

The second finding took three timescales to see, and they do not agree — which is the point.

TIMESCALE	WHAT WAS MEASURED	A POISSON PROCESS GIVES
1 second	variance/mean of the count = 3.90 , up to 20.1	1.0
50 ms	P(gap < 50 ms) = 1.03x the exponential	1.0
10 ms	P(gap < 10 ms) = 1.97x , up to 4.0	1.0
50 ms	branching ratio = 0.02	0

Read together:

- **At 50 ms the arrivals are exactly Poisson**, and a branching ratio of 0.02 rules out a cascade. One crackle does not make the next one more likely — which is worth knowing, because “crackles come in bursts” is the obvious thing to assume and it is measurably wrong at that scale.
- **Below 10 ms there are twice as many gaps as Poisson allows**. A crackle is not one impulse. It is a short train of one to three, inside ten milliseconds, and then nothing: gas breaking out of a split in the wood in stages rather than all at once. That is *Burst*.
- **At one second the count varies four to twenty times more than its mean**. The *rate itself* wanders, on the same seconds-long timescale as the flame’s own surging. That is *Flare*, and once the rate is allowed to wander the whole excess is accounted for without anything else being bursty.

So the spawner has three levels and no cascade: a slowly wandering rate, a Poisson process at that instantaneous rate, and a one-to-three pulse train per arrival.

What is in this manual

The chapters below cover installing it, getting a first sound, how the model works and why it is built the way it is, the window, then a generated reference for every parameter and every factory preset. The last two chapters are honest about what does not fit yet and where the model comes from.

Installing

CrackleBlaze is one self-contained `.clap` file plus its `presets` folder. There is no installer, no shared library to place and nothing to register.

Linux

CLAP hosts scan `~/ .clap`. Copy the plugin folder from the release archive into it so that you end up with:

```
~/ .clap/CrackleBlaze/CrackleBlaze.clap
~/ .clap/CrackleBlaze/presets/
```

Then rescan plugins in your host. X11 and Cairo are the only external dependencies of the plugin window, and any Linux machine that can run a DAW has both.

Windows

CLAP hosts scan the common CLAP folder. Copy the plugin folder into:

```
C:\Program Files\Common Files\CLAP\CrackleBlaze\
```

so that `CrackleBlaze.clap` and `presets\` sit inside it, and rescan. There are no DLLs to install beside it — the window and its Cairo are linked in.

Keep the plugin and its presets together. The plugin finds its factory presets by looking for a `presets` directory next to its own binary. Moving the `.clap` file on its own leaves it with no factory presets at all.

Hosts

Tested with Bitwig Studio and Reaper. In Bitwig, `~/ .clap` is scanned by default; after installing, restart it or rescan under *Settings* → *Locations* → *Plug-in Locations*. CrackleBlaze then appears as an instrument under the vendor *Ravetracer*, and the factory presets are indexed through CLAP's preset discovery, so they show up in the host's own browser as well as in the plugin window.

Your own presets

SAVE in the plugin window writes to a user preset folder, which is created on first use:

PLATFORM	LOCATION
Linux	<code>\$XDG_CONFIG_HOME/CrackleBlaze/presets</code> , or <code>~/ .config/CrackleBlaze/presets</code>
Windows	<code>%APPDATA%\CrackleBlaze\presets</code>

User presets appear in the browser immediately, listed after the factory ones. They are plain text, so they can be copied between machines, kept in version control, or edited in any editor.

Building from source

The suite builds with a C++17 compiler and CMake 3.16 or newer. From the plugin folder:

```
./install.sh
```

That configures, builds, runs the self-test and installs to `~/clap/CrackleBlaze/`. `CrackleBlaze_PREFIX=/some/where`
`./install.sh` overrides the destination, and `-DCLAP_INCLUDE_DIR=/path/to/clap/include` points the build at a CLAP
checkout it cannot find by itself.

A first sound

1. Load CrackleBlaze on an instrument track.
2. Draw a long note — eight bars or more. The fire burns for as long as the note is held, and the release fades it out.
3. Open the preset browser by clicking the preset name in the window header and try **Cottage Hearth**, then **Dry Kindling**, then **Furnace Roar**.

Those three are the span of the instrument. The middle one is almost nothing but countable ticks; the last is almost nothing but bed.

Three things to try first

Change the fire. *Fire* selects one of five colours, and they are not names someone invented for a menu: they are the five clusters the reference library's *beds* fall into, each shipped as the measured octave-band centroid of its members. *Blend* crossfades to the next one, so the five are a continuum rather than five settings. Deep Blaze peaks at 125 Hz and falls 15 dB by 8 kHz; Stove Draught is 22 dB down at 1 kHz and peaks at the very top of the spectrum.

Move Sap. This is the wet/dry axis, and it is the widest per-recording variable in the whole library: from 0.03 in a big open blaze to 0.57 in a fireplace full of unseasoned wood. At the bottom the fire is all sharp ticks; at the top most of what you hear is steam leaving the wood, which takes seven times as long to die away. Nothing else changes — the two populations have the same measured colour.

Move the fire away. Raise *Distance*. Air absorption takes the top off, the individual crackles multiply and shrink, their edges smear, and at the far end nothing is countable any more. At zero the air filter is bypassed outright rather than parked out of the way, because no distance means no air to absorb.

How it works

The roar, and where its colour comes from

The bed is one noise source per channel through eight bandpasses at octave centres from 125 Hz to 16 kHz, and the gains are **solved** rather than set: an octave-wide bandpass leaks into the octave beside it, so driving each band at its target leaves the sum wrong. The solver asks instead for the gains whose *summed* response is the measured curve.

The five colours are the five clusters `tools/analysis/shapes.py` finds in the library, each shipped as its cluster's centroid. A table of eight numbers is a formula and not a sample; nothing here reproduces recorded audio.

The crackles had to be gated out before that clustering meant anything. At a median crest factor of 31.7 dB, the plain spectrum of a fire is substantially the spectrum of its crackles — so clustering on it sorts the recordings by how close the microphone was, not by what kind of fire they are. `tools/analysis/bed.py` drops the loudest quarter of every recording's 4 ms frames first. What the crackles add back is measurable and is exactly where you would expect:

	125 HZ	250	500	1K	2K	4K	8K	16K
the crackles add	-7.6	-4.0	-1.8	+0.4	+4.0	+5.9	+3.8	+1.0

A fire flares as one flame

This is the sharpest difference between this plugin's bed and RiverFlow's, and it is a single measurement. The correlation between one octave band's 100 ms envelope and its neighbour's is:

	MEASURED
CrackleBlaze's library	0.54 ... 0.91, median 0.85
RiverFlow's library	0.07 ... 0.25

A river's bands wander independently — it is a great many small events at different scales, and the low end surging tells you nothing about what the hiss is doing. A fire is one object. When it flares, all of it flares.

So the model is one common random walk at $\sqrt{0.85}$ plus a small independent walk per band at $\sqrt{0.15}$, which reproduces the measured 0.85 exactly. And the same common walk drives the crackle rate, because in a real fire a flare-up is both more roar and more crackling — that sharing is what produces the measured variance-to-mean ratio of 3.9 without any part of the spawner being bursty.

It is a filtered random walk and **not** an oscillator. The envelope spectrum of every reference falls smoothly at about -5.7 dB per decade over 0.1–10 Hz, and the strongest frequency in it ranges from 0.12 to 2.54 Hz with no peak that survives from one recording to the next. A fire has no rhythm. *Flare Rate* sets the corner of the noise that drives the walk, not a frequency you will hear.

A crackle is a click, not a ring

It is tempting to model a crackle as a struck piece of wood — a resonator with a pitch. The measurement says no.

For every detected crackle, `tools/analysis/crackshape.py` takes 8 ms from the onset, subtracts the spectrum of the 8 ms before it, and averages the excess over every event in the recording. What is left is the crackle alone, with the roar it sat in removed. Its **spectral flatness is 0.67** across the library, where 1.0 is white noise and a resonance would be a small fraction. There is no ring in it. The few recordings that measure low (0.15–0.19) are the ones whose event spectra are dominated by bed leakage at the bottom, not by a resonance.

So a crackle here is a shaped noise burst with a fast attack and an exponential decay, and nothing else. Giving it a resonator would be inventing a physical object the recordings do not contain.

What it *does* have is a low end: the measured event spectrum is still within 8 dB of its peak at 250 Hz, which is *Body*. Without that a crackle is all fizz and no wood.

MEASURED ACROSS THE LIBRARY	
rate	11.5 ... 51.0 /s, median 29
prominence over the roar	6.5 ... 20.2 dB, median 13.2
amplitude spread	log-normal, σ = 6.0 dB
decay to -10 dB	median 2.5 ms (1.0 ... 5.5)
spectral flatness	median 0.67

A sizzle is a crackle held open

The population is bimodal in duration, and only in duration. Splitting the detected events at 8 ms and measuring each half separately:

	SHARE	DECAY TO -10 DB	SPECTRUM
ticks	0.79	2.5 ms	0 / -2.2 / -6.6 / -6.4 / -7.9 / -11.0 / -18.3 dB
hisses	0.21	19 ms	0 / -3.6 / -3.7 / -5.5 / -7.0 / -7.9 / -20.5 dB

Seven to eight times the decay, and the same colour to within about 2 dB per octave. These are not two different kinds of event. They are the same event held open for different lengths of time, because steam has to leave the wood and a bursting pocket of dry gas does not.

That is why *Sap* is a single control that moves events from one population to the other, rather than two independent layers — and why it is a parameter at all rather than a constant. Its measured range across the library is 0.03 to 0.57, which is the widest spread of any quantity measured here.

Steam adds a narrow, high, per-event jet on top of the sizzle’s broadband noise: the whistle of steam leaving a split. That one is modelled from the physics rather than fitted, because nothing in the library isolates it.

A log giving way

The fourth layer is rare and easy to miss. Finding it at all needs an onset detector run on 80–300 Hz with anything that *also* jumps in the top thrown away, because a loud crackle leaks into every band and would otherwise be counted twice. What survives:

MEASURED	
rate	0 ... 136 a minute, median 6
prominence over the low bed	13.9 ... 25.7 dB, median 16.1
decay	6 ... 10 ms

A thump, not a boom, and the only layer in the plugin with no top end at all. Two of the seventeen usable recordings have none. *Log Collapse* is the one factory preset that puts it where you cannot miss it.

Nothing below 60 Hz

Eight of the twenty-five reference recordings carry between 27 and 60 per cent of their total energy below 60 Hz. None of it is fire.

`tools/analysis/lowend.py` correlates each recording’s 20–60 Hz envelope against its 1–4 kHz envelope. In twenty-four of the twenty-five the correlation is under 0.2, and in half of them it is indistinguishable from zero — while the one exception carries only 2.4 per cent of its energy down there. It is traffic, ventilation, wind and handling noise on the microphone. Synthesising it would be fitting the recordist’s afternoon.

So nothing below 60 Hz is generated, *Highpass* defaults there, and every analysis script high-passes before measuring anything.

Eight recordings were discarded outright on the same evidence. Above 800 Hz they carry a flat -45 dB plateau out to 12 kHz — a dither floor, not content — and their crest factors of 11.8 to 16.5 dB say the same thing: nothing impulsive is in them. They are rumble beds with a fire somewhere underneath. Seventeen references carry the whole fit.

Distance and the hearth

Distance does what distance does outdoors, all at once: air absorption takes the top off, what is left tilts downwards, the individual events multiply and shrink — a fire far enough away is a bonfire, not a hearth heard quietly — and their edges smear with the air they crossed. Energy is held roughly constant across that, so *Distance* is a character control and not a volume control.

Hearth sets how much early reflected field there is. *Open* has almost none: a fire in a field has nothing close enough to reflect off, and eight discrete early reflections outdoors is exactly what makes a reverb sound like a bathroom. *Fireplace* and *Stove* have a great deal — the two references recorded inside a fireplace measure as the most reverberant in the library, and *Open Fireplace* is fitted to them.

The plugin window

CrackleBlaze draws its own window with X11 and Cairo on Linux, and Cairo on the win32 backend on Windows. There is no toolkit dependency, which is what keeps the plugin one self-contained file. It is embedded in the host's own window through CLAP's GUI extension and repainted from the host's timer, and it resizes by zooming one layout, from half size to four times.

The window is the suite's, in CrackleBlaze's own ember-orange theme, with tongues of flame along the bottom of the header behind the wordmark and an ember thrown up wherever a crackle happens. The embers are deterministic in the crackle number, so a pinned *Random Seed* gives a repeatable picture as well as a repeatable sound.

Gestures

GESTURE	EFFECT
Drag a knob up or down	Change the value
Shift-drag	Fine control, a fifth of the travel
Double-click a knob, or right-click anything	Back to the default
Scroll wheel over a control	Step the value
Click a knob's value	Type a value in, with units
Click a selector's name	Open its list and pick a value
Click a selector's ◀ or ▶	Step one choice
Click the preset name	Open the preset browser
◀ or ▶ beside the preset name	Previous or next preset
SAVE	Save the current settings as a user preset
MIXER	Open the layer mixer

The line at the bottom of the window shows what the control under the pointer does — the same one-line explanation that appears in the *What it does* column of the parameter reference.

The mixer

Every layer's level lives on the panel that layer belongs to, which is right for editing one layer and wrong for balancing them against each other. **MIXER** opens the same controls arranged the other way round: one strip per layer, its level as a fader, its width as a slim slider under it. They are the plugin's own parameters, not a second set — moving a fader moves the knob on the panel, and the host sees the same automation it always would.

M mutes a layer and **S** solos one, which is how a preset gets built from nothing: solo one layer, get it right, bring the next one back. Several layers can be soloed at once.

Mute and solo are **not** parameters and are not saved with a preset. They work by holding a layer's level at the bottom of its range and remembering what it was, so while either is active the preset bar shows a **SOLO ON** or **MUTE ON** chip — click it to release every hold at once. **SAVE** releases them first, so a muted layer can never be written into a preset as a silent one, and so does loading a preset. Touching a held layer's fader also releases the holds rather than fighting the hand on it. Closing the mixer does not: soloing a layer and then going to its knobs on the panels is what the mixer is for.

Parameter reference

Every range, default, unit and explanation below is read from the plugin's own parameter table when this manual is built, so it cannot drift from what the window and the host show.

There are 47 parameters, grouped the way the plugin window groups them. Every one is automatable and modulatable from the host, and every one can be typed in directly by clicking its value in the window. The name in `code` is the key used in preset files.

BLAZE

PARAMETER	RANGE	DEFAULT	WHAT IT DOES
Fire <code>fire_type</code>	Deep Blaze, Log Fire, Camp Fire, Open Flame, Stove Draught	Log Fire	Which measured colour the roar has. The five choices are the five clusters the reference library's <i>beds</i> fall into, each shipped as its cluster's measured octave-band centroid – from Deep Blaze, which peaks at 125 Hz and falls 15 dB by 8 kHz, to Stove Draught, which is 22 dB down at 1 kHz and peaks at the very top. Gating the crackles out before clustering is what makes these five shapes mean anything. The library's median crest factor is 31.7 dB, so the ungated spectrum of a fire is largely the spectrum of its crackles, and clustering on that sorts recordings by how close the microphone was rather than by what kind of fire it was.
Blend <code>fire_blend</code>	0.0 % ... 100.0 %	0.0 %	Crossfades from the chosen colour towards the next one, so the five measured shapes are a continuum rather than five settings. At 0 the roar is exactly the measured centroid.
Roar Level <code>roar_level</code>	-inf dB ... 6.00 dB	-31.00 dB	Level of the combustion roar, and it reads as exactly that: the roar is normalised to unit RMS before this is applied, whichever colour is selected, so -26 dB is a roar at -26 dBFS and changing colour is not also a level change. A fire is much less bed than a river is. The library's crest factor is 31.7 dB against a river's 19.6 – so where a river is mostly its bed with events on top, a fire is a quiet roar with very loud, very short things happening over it.
Tilt <code>roar_tilt</code>	-100.0 % ... 100.0 %	0.0 %	Tilts the whole measured shape brighter or darker, +/-6 dB/octave about 1 kHz. Moves a colour off its measured centre without discarding its shape.
Body <code>roar_body</code>	0.0 % ... 100.0 %	50.0 %	Weight of the roar's lowest two bands, 125 and 250 Hz. This is where a fire's size lives: the library's bed measures a median of -4.0 dB at 125 Hz against -14.9 dB at 2 kHz, so the roar is a low-frequency phenomenon whatever the crackles on top of it are doing.
Flare <code>flare</code>	0.0 % ... 100.0 %	55.0 %	How much the fire surges – and it surges as <i>one fire</i> , which is the measured difference between this plugin's bed and RiverFlow's. The correlation between one octave band's 100 ms envelope and its neighbour's is 0.54 to 0.91, median 0.85, where a river measures 0.07 to 0.25. So a single common walk drives every band, with only a small independent part on each; a river needs eight separate ones. Fitted to the bed's low bands, where the measured CV is 0.39 at 125 Hz and 0.45 at 250. The upper-band figures run to 1.28, but those still carry crackle and sizzle leakage that no gate removes cleanly, so they are not what this was set from. It moves the crackle rate as well as the roar's level, because in a real fire a flare-up is both – see Crackle Rate.
Flare Rate <code>flare_rate</code>	0.05 Hz ... 5.00 Hz	0.55 Hz	How fast that surging is. It is a filtered random walk rather than an oscillator: the envelope spectrum of every reference falls smoothly at about -5.7 dB per decade over 0.1-10 Hz, and the strongest frequency in it ranges from 0.12 to 2.54 Hz with no peak that survives from one recording to the next. A fire has no rhythm. This sets the corner of the noise that drives it.
Draught <code>draught</code>	0.0 % ... 100.0 %	15.0 %	Air being pulled into the fire: a high-frequency hiss added over the roar, steadier than the roar itself. It is what separates a stove with its lid open from the same fire in a room, and Stove Draught is the shape fitted to that reference.

CRACKLE

PARAMETER	RANGE	DEFAULT	WHAT IT DOES
Crackle Rate <code>crackle_rate</code>	0.50 /s ... 120 /s	29.0 /s	Crackles a second, averaged. Measured 11.5 to 51.0 across the library, median 29. Averaged, because the rate does not hold still. The variance of the count in a one-second window is 3.9 times its mean across the library and up to 20 times in places, where a Poisson process would give exactly 1. That whole excess is the rate itself wandering on a seconds-long timescale – the same walk that drives Flare – and not the crackles clustering: at 50 ms the arrivals are Poisson to within 3%, and the branching ratio is 0.02, which

rules out one crackle provoking the next.

Crackle Level <code>crackle_level</code>	-inf dB ... 6.00 dB	-16.00 dB	Level of the crackles. Measured as prominence over the bed they sit on: 6.5 to 20.2 dB, median 13.2.
Decay <code>crackle_decay</code>	0.30 ms ... 40.0 ms	1.80 ms	How long one crackle takes to fall 10 dB. Measured median 2.5 ms, from 1.0 to 5.5 – these are clicks, and at the short end of the range a crackle is over before the ear can hear a pitch in it.
Tone <code>crackle_tone</code>	400 Hz ... 12.00 kHz	4.00 kHz	Where the crackle's energy sits. The measured spectrum, with the roar underneath subtracted, falls about 2 dB per octave from 250 Hz and then drops away above 8 kHz; the crackles add +5.9 dB to the library's median spectrum at 4 kHz, +4.0 at 2 kHz and +3.8 at 8 kHz. This moves that whole shape.
Spread <code>crackle_spread</code>	0.00 dB ... 18.00 dB	6.00 dB	How unequal the crackles are. Their amplitudes are log-normal with a measured standard deviation of 0.30 decades, which is 6 dB – so the loudest crackle in a few seconds is some 15 dB over the quietest, and a fire with this at zero sounds mechanical immediately.
Burst <code>burst</code>	1.00 ... 4.00	2.00	Pulses per crackle, averaged. A crackle is not one impulse: below 10 ms the references carry 1.97 times as many intervals as a Poisson process allows, and up to 4 times in places, while at 50 ms they carry exactly as many. Something happens two or three times inside ten milliseconds and then does not happen again – a pocket of gas breaking out of the wood in stages rather than at once.
Snap <code>snap</code>	0.0 % ... 100.0 %	60.0 %	How sharp the front of a crackle is. At the top it is an impulse with the decay hung off it; lower, the energy is spread into the first milliseconds and the crackle reads as a thud rather than a tick.
Body <code>crackle_body</code>	0.0 % ... 100.0 %	30.0 %	How much low end each crackle carries, below its Tone. The measured crackle spectrum does not stop at the bottom – it is still within 8 dB of its peak at 250 Hz – and without this a crackle is all fizz and no wood.

SIZZLE

PARAMETER	RANGE	DEFAULT	WHAT IT DOES
Sap <code>sap</code>	0.0 % ... 100.0 %	21.0 %	What fraction of the events are sizzles rather than crackles – wet wood rather than dry. This is the widest per-recording variable in the whole library: measured from 0.03 in the big open blaze to 0.57 in <code>fire_near_open_close</code> , median 0.21. The two populations are the same event held open for different lengths of time. Their measured spectra agree to within 2 dB per octave; what differs is that a sizzle takes 19 ms to fall 10 dB where a crackle takes 2.5. Steam has to leave the wood, and that takes time a bursting gas pocket does not need.
Sizzle Level <code>sizzle_level</code>	-inf dB ... 6.00 dB	-23.00 dB	Level of the sizzles.
Decay <code>sizzle_decay</code>	5.00 ms ... 200 ms	19.0 ms	How long a sizzle takes to fall 10 dB. Measured median 19 ms, from 11 to 30 – seven to eight times a crackle's.
Tone <code>sizzle_tone</code>	400 Hz ... 12.00 kHz	4.50 kHz	Where a sizzle's energy sits. Defaults slightly above Crackle Tone because the measured sizzle spectrum is a little flatter across the upper middle, but the two are close on purpose: the measurement says these events differ in duration, not in colour.
Steam <code>steam</code>	0.0 % ... 100.0 %	25.0 %	How much of a sizzle is a narrow jet rather than broadband noise. Steam leaving a split in the wood whistles; this decides how much of that is in the sound.

SETTLE

PARAMETER	RANGE	DEFAULT	WHAT IT DOES
Settle Rate	0.01 /s	0.10 /s	How often a log gives way. Rare on purpose: measured a median of six a minute across the library, and zero

Settle Rate settle_rate	... 2.00 /s		in two of the seventeen recordings. Finding them at all needs an onset detector run on 80-300 Hz with anything that also jumps in the top thrown away, because a loud crackle leaks into every band and would otherwise be counted twice.
Settle Level settle_level	-inf dB ... 6.00 dB	-25.00 dB	Level of the settles. Measured 13.9 to 25.7 dB over the low bed, median 16.1.
Tone settle_tone	40.0 Hz ... 400 Hz	150 Hz	Where the thump sits. The population was measured in the 80-300 Hz band and has no top end at all – that is what distinguishes it from a loud crackle.
Decay settle_decay	3.00 ms ... 300 ms	8.00 ms	How long a settle takes to fall 10 dB. Measured 6 to 10 ms – a thump, not a boom. Longer than that and it stops being wood and starts being a drum.

HEARTH

PARAMETER	RANGE	DEFAULT	WHAT IT DOES
Hearth hearth_type	Open, Fire Ring, Stone Hearth, Fireplace, Stove, Cavern	Fireplace	What the fire is burning in, which sets how much early reflected field there is. Open has almost none – a fire in a field has nothing close enough to reflect off, and eight discrete early reflections outdoors is exactly what makes a reverb sound like a bathroom. A fireplace and a stove have a great deal, and the two references recorded inside one measure as the most reverberant in the library.
Distance distance	0.0 % ... 100.0 %	15.0 %	How far away the fire is. Air absorption plus a downward tilt, with the events multiplied and their edges smeared – and bypassed outright at zero, because no distance means no air to absorb.
Air air	0.0 % ... 100.0 %	50.0 %	How much of the top end that distance costs. Humid air absorbs less than dry.
Width width	0.0 % ... 100.0 %	70.0 %	How widely the crackles, sizzles and settles are spread across the stereo field. The library measures an L/R correlation from -0.52 to 1.00, median 0.69.
Roar Width roar_width	0.0 % ... 100.0 %	45.0 %	The same for the roar, separately. A fire's roar is one turbulent source and decorrelates less than the discrete events around it.
Space space_amount	0.0 % ... 100.0 %	25.0 %	How much reverberant field there is beyond the early reflections.
Size space_size	0.0 % ... 100.0 %	40.0 %	How big that space is.
Damping space_damping	0.0 % ... 100.0 %	60.0 %	How quickly the space eats the top end. Brick and stone eat a great deal.

FILTER

PARAMETER	RANGE	DEFAULT	WHAT IT DOES
Filter Type filter_type	Lowpass, Bandpass, Highpass, Notch	Lowpass	Shape of the output filter.
Highpass highpass	20.0 Hz ... 2.00 kHz	60.0 Hz	Permanent highpass. The default of 60 Hz is measured, not cautious: eight of the twenty-five references carry between 27 and 60 per cent of their energy below 60 Hz, and in twenty-four of the twenty-five that energy's envelope is uncorrelated with the fire above it ($ r < 0.2$). It is traffic, ventilation and handling noise on the microphone, not fire, so the engine does not synthesise it.
Filter Cutoff filter_cutoff	30.0 Hz ... 20.00 kHz	20.00 kHz	Corner of the output filter.
Filter Resonance filter_reso	0.0 % ... 100.0 %	10.0 %	Resonance of the output filter.

Filter Mix filter_mix	0.0 %	0.0 %	How much the filter follows the input level.
---------------------------------	-------	-------	--

Filter Key Track	0.0 % ...	0.0 %	How far the filter follows the played note.
<code>filter_keytrack</code>	100.0 %		

ENVELOPE

PARAMETER	RANGE	DEFAULT	WHAT IT DOES
Attack <code>attack</code>	1.00 ms ... 20.00 s	800 ms	Fade-in of the whole fire when a note starts.
Decay <code>decay</code>	5.00 ms ... 20.00 s	499 ms	Fall from the initial level to the sustain.
Sustain <code>sustain</code>	0.0 % ... 100.0 %	100.0 %	Level held while the note is down.
Release <code>release</code>	5.00 ms ... 30.00 s	1.50 s	Fade-out after the note is let go.
Velocity To Level <code>vel_to_level</code>	0.0 % ... 100.0 %	40.0 %	How much velocity sets the level.
Velocity To Fire <code>vel_to_fire</code>	0.0 % ... 100.0 %	30.0 %	How much velocity sets how much fire there is: the event rates and the roar's brightness together, which is what a bigger fire actually does.

OUTPUT

PARAMETER	RANGE	DEFAULT	WHAT IT DOES
Output Gain <code>gain</code>	-inf dB ... 12.00 dB	-3.00 dB	Output level. The default leaves 3 dB of headroom, which a fire needs more than a river does: its amplitudes are log-normal with a measured 6 dB spread, so the loudest crackle in a minute stands far above the one the meter usually shows.
Max Events <code>max_events</code>	16 ... 2048	1024	Ceiling on crackles, sizzles and settles alive at once, so the cost is bounded. Higher here than in the suite's other plugins because a fire spawns far more events a second than a river does.
Random Seed <code>seed</code>	0 ... 999	0	Non-zero gives the same fire every time. Zero is always different.

The preset library

Generated from the preset files themselves.

18 factory presets ship with the plugin. They are installed beside the binary and exposed through CLAP preset discovery, so they appear in the host's own browser as well as in the plugin window's.

Big Blaze

A large open fire a few metres back. Deep Blaze carries the most low end of the five measured beds and the dullest top, which is what a big fire heard from a distance actually is: the crackles that survive the air are the loud ones. (*large, deep, outdoor*)

Bonfire

A bonfire at ten paces, wide across the field of view. Camp Fire's bed with more body under it and a rate near the top of the library. (*bonfire, outdoor, wide*)

Burning Roof

A building alight: flames stacked metres high, heard from across the street. The roar is most of it, the crackle rate is at the top of the measured range, and the settles are frequent because a structure fire is continually collapsing. (*large, structure, roaring*)

Camp Fire

The default patch, and the one the engine was fitted at. A camp fire in the open at arm's length: the measured Camp Fire bed, a fire ring with almost nothing to reflect off, and crackles standing the library's median 13 dB over the roar. (*camp, outdoor, default*)

Cottage Hearth

A small fire in a cottage fireplace. Log Fire is the library's commonest bed shape and this is what it was named for: a strong 125 Hz, a scooped middle, and the top coming back for the crackles. (*hearth, indoor, cosy*)

Crackle Close

The microphone on top of the fire. No distance at all, so nothing is smeared and the crackles stand at the top of the measured prominence range – 20 dB over the roar, which is `fire_near` and `fireplace-at-castle-mountain-bungalows`. (*close, detailed, dry*)

Distant Fire

The same fire, across a field. Distance multiplies the events and divides them, smears their edges and takes the top off with the air they crossed – so a fire far enough away stops being countable ticks and becomes a texture. (*distant, soft, air*)

Dry Kindling

Dry sticks catching. Sap at the bottom of the measured range, the rate at the top, and Snap as far up as it goes – these are ticks with nothing behind them. (*kindling, sharp, fast*)

Dying Embers

Almost out. The roar has gone, the sharp ticks with it, and what is left is the occasional sizzle out of a hot log – Sap near the top of its measured range, because what is still evaporating is all that is still making sound. (*embers, quiet, sparse*)

Forest Campfire

A camp fire among trees at night. A little more space than the open, because a forest does return something – but nothing like a room. (*camp, forest, night*)

Furnace Roar

Almost all bed. A furnace in a vaulted space, where the roar and its early field swamp the individual crackles – the far end of the plugin's range from Dry Kindling. (*furnace, roar, cavern*)

Hearth Glow

A settled fire in a stone hearth, heard from a chair across the room. Quiet, slow and low: the bed does most of the work and the crackles are punctuation. (*hearth, quiet, warm*)

Log Collapse

A fire that keeps giving way. The settle layer is rare in the library – a median of six a minute – and this is the one preset that puts it where you cannot miss it, at the top of the measured range. (*settle, thumps, shifting*)

Open Fireplace

Fitted against `fire-in-fireplace_close-up_reverberant`, the most reverberant pair in the library: the microphone inside the opening, so the early field is enormous and the 125 Hz band is 10 dB down on everything above it. (*fireplace, reverberant, close*)

Sauna Stove

A sauna stove: an enclosed firebox with a deep body and a hard, bright early field off the metal and the stones. (*stove, enclosed, deep*)

Slow Burn

A fire barely being fed. Everything slow: a low rate, a long flare and sizzles held near the top of their measured decay. (*slow, sparse, long*)

Wet Wood

Wood that has not dried. Sap at the top of the measured range: 57 per cent of the events are sizzles rather than ticks, which is `fire_near_open_close` – the wettest recording in the library. (*sizzle, steam, damp*)

Wood Stove

The lid open on a wood stove. Stove Draught is the one shape in the library with no members but itself – 22 dB down at 1 kHz and peaking at 16 – and the draught pulling air through it is most of what is heard. (*stove, draught, bright*)

Using it in a host

Notes

A note lights the fire and holding it keeps it burning. The ADSR shapes the whole scene rather than individual events, and *Release* is what fades it out. *Velocity To Fire* makes velocity behave like a bigger fire: it raises the event rates and the roar's weight together, which is what feeding a fire actually does.

Fire needs long notes. Its character is in the statistics of many events and in the slow wander of the flare, and neither is audible in a short one.

There are 8 voices, each with its own roar, and 2048 events in a shared pool, so the number of held notes cannot multiply the cost without bound. The event pool is larger than elsewhere in the suite because a fire spawns far more events a second than a river does — up to 51 crackles a second, each of them a burst of two or three.

Automation and modulation

Every parameter is automatable, and host parameter modulation (`CLAP_EVENT_PARAM_MOD`) is supported globally. Per-note modulation and note expressions are ignored. Parameter changes are sample accurate, so an automation ramp over a bar — a fire catching over *Crackle Rate* and *Roar Level* — is a smooth move rather than a staircase at block boundaries.

Reproducible renders

Random Seed at 0 never repeats. Any other value renders the same fire for the same notes every time, and the output is byte-identical across runs at both 48 and 96 kHz. Each voice's noise generators and its first event timers are derived from its own slot and key rather than from the shared generator, so a fixed seed promises the same fire even when note events arrive before the seed has been applied.

CPU cost

Cost is bounded by the pools. The roar is the fixed part — sixteen bandpasses per voice per sample — and it does not vary with any parameter. What varies is *Crackle Rate* times *Burst*, which decides how many events are running, *Sizzle Decay*, which decides how long each one lives, and *Space*, which is the room model. At the default settings roughly two events are alive at any moment.

Rendering and tails

Fire has a very short tail compared with surf or thunder — the longest thing still to come after a note is the release plus the room, since even a sizzle is gone in a tenth of a second. When bouncing, a second or two past the end of the note is enough.

Headroom

Every factory preset carries a trim in its own output gain, measured rather than judged: each was rendered 24 dB down with the seed pinned, its true peak read off, and its gain set so that peak lands at -3 dBFS — then capped at -3 dB, so a preset that is meant to be quiet stays quiet instead of being normalised up to meet the others.

That cap matters more here than elsewhere in the suite. A fire's amplitudes are log-normal with a measured 6 dB spread, so the loudest crackle in half a minute stands some 18 dB over the median one. Without the trim, several presets rendered into the output stage's soft clipper — and a clipped crackle is a crackle with its transient taken off, which is the one thing this plugin must not do.

What is still being fitted

CrackleBlaze 0.1.0 is a first fit, and the measurements say where it is not there yet. All of this is in the plugin's `TODO.md` with what is known about it.

Nothing has been validated by ear. Every number in this manual is a measurement, and the suite's own rule is that a number agreeing with a number proves nothing about the sound. This is the first thing on the list.

The default patch is not varied enough. It renders with an envelope variation of 0.56 at 4 ms against the library's 0.85, and 0.41 at 50 ms against 0.56. The crest factor is 29.8 dB against a median of 31.7. The fire is a little too even, minute to minute, and the likeliest cause is the next item.

The flare depth above 500 Hz is extrapolated, not measured. The bed's envelope variation was measured cleanly at 125, 250 and 500 Hz (0.39, 0.45, 0.57). Above that no gate separates the bed from the crackle and sizzle layers riding on it, and since this engine generates those separately, shipping the measured 1.28 at 2 kHz would make the bed wobble by an amount that in the reference *was* the crackles — and then add the crackles on top. The upper five bands therefore continue the trend and flatten. Fitting them properly means fitting the render's own per-band variation against the library rather than the bed's.

The crackles are not unequal enough. The rendered amplitude spread measures 4.2 dB against the library's 6.1, even though the parameter is set to the measured 6.0. Something in the chain between the draw and the detector is narrowing the distribution.

The crackles are a little bright. The rendered event centroid is 3.7 kHz against a library median of 3.0.

The bed's octave shape is within about 1 dB, not exact. The bandpasses are set to a Q of 2.2, which was chosen by modelling: at the octave-wide Q of 1.4 the bank could not reach the contrast two of the five shapes ask for and left them 3 to 4 dB short, and at 2.8 it is exact but the ripple between band centres reaches 2.8 dB peak-to-peak. 2.2 is where the error stops mattering before the ripple starts to.

The hearths and the space controls are chosen, not measured. A reverberation measurement needs an impulse and a known source position, and the library has neither.

Eight of the twenty-five references were discarded, for the measured reasons in *Nothing below 60 Hz*. A larger library — particularly of large open fires, where only a handful are usable — would narrow every number here.

Reference and limits

Where the model comes from

- **The reference library itself**, which supplied nearly everything: the five octave-band colours, the flare's depth and its correlation across bands, the crackle rate, amplitude law, decay and spectrum, the tick/hiss split, the settle population, and the three timescales the spawner is built on. The numbers and the scripts that produced them are in the plugin's `tools/analysis/`.
- The physical reading behind the layers is the standard one: **combustion roar** as broadband low-frequency noise radiated by unsteady heat release in a turbulent flame, which is why the roar is a low-frequency phenomenon whatever the crackles above it do; and **crackling** as pyrolysis gas breaking out of heated wood, which is why it is impulsive and why it has no pitch. The library decided every number; the physics only decided what to measure.
- Where nothing in the library isolates a quantity — *Steam*, the hearth enclosures, the space controls — the model is stated as chosen rather than fitted, both here and in the chapter above.

What it does not do

- **No rain, no wind, no wildlife.** Those are other instruments in the suite.
- **No sub-bass.** Nothing below 60 Hz is generated, for the measured reason above. If you want the low rumble a large fire puts through a floor, that is not in the references and is not in here.
- **No pitched resonance in the crackles.** Measured flatness says there is none to model.
- **No per-note modulation.** Global parameter modulation only.
- **Nothing above 16 kHz is fitted.** The topmost measured octave straddles the limits of the reference recordings themselves.