



VERDALIS
Natural sound synthesis

RainyDay

Synthesised rain

Version 1.10.1

CLAP instrument for Linux and Windows

Part of the Verdalis Plugin Suite

2026-10-04

Contents

Introduction

- 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

- Droplet arrival
- A single droplet
- Droplet size follows from physics
- Trickle: drops landing on a surface
- The bed
- Space and distance
- Level behaviour

The plugin window

- Gestures
- The mixer
- Saving a preset

Parameter reference

- Output
- Rain
- Distant
- Close
- Space
- Filter
- Envelope
- System
- Trickle

The preset library

Using it in a host

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

Reference and limits

- Where the model comes from
- Notes and limits
- Versioning
- License

Introduction

RainyDay is a CLAP instrument that generates rain. It contains no samples and loads no audio files: every droplet, every splash and the whole background wash are computed from noise, oscillators and filters while the plugin plays. Two instances never produce the same rain, and one instance never repeats itself.

That is the point of it. Recorded rain arrives with its distance, its impact surface and its room already fixed in the file, and however long the recording is, the ear eventually finds the place where it comes round again. A synthesised rain field has no loop point to find, and the things a recording has baked in are separate controls: what the rain is falling on, how far away it is, how much high end the air between you and it has taken, and the space the whole field sits in.

The engine is not a noise generator with filters on it. Droplets arrive as a statistical process, each one is given a size, and its pitch, its level and its ring time all follow from that size the way they follow in physics. The individually audible drops are synthesised one at a time; the ones too distant to resolve are summed statistically into a bed, because synthesising ten thousand inaudible drops a second would cost a great deal and sound the same. *How it works* describes the whole chain.

The parameter values and the factory presets were not dialled in by ear alone. Both were fitted against reference recordings by measuring the recording and the plugin's own output with the same analysis code and moving values until the two agreed. Where a number in this manual comes from a measurement or a paper, it says so.

What is in this manual

Installing and *A first sound* are enough to get rain out of the plugin. *How it works* explains the model, which is worth reading before spending time in the parameter list — most of the controls make immediate sense once the model does. *Parameter reference* and *The preset library* are generated from the plugin itself, so they always describe the version on the cover. *Using it in a host* covers notes, automation, reproducible renders and CPU cost.

Installing

RainyDay 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/RainyDay/RainyDay.clap
~/ .clap/RainyDay/presets/
```

Then rescan plugins in your host. Nothing else is needed: X11 and Cairo are the only external dependencies of the plugin window, and any Linux machine that can run a DAW already has both.

Windows

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

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

so that `RainyDay.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*. RainyDay 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/RainyDay/presets</code> , or <code>~/ .config/RainyDay/presets</code>
Windows	<code>%APPDATA%\RainyDay\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.

COLLECTIONS

A preset library is a shelf, and a long one is worth tidying. Your own presets are filed in **collections**: folders one level under the user preset directory. The browser lists the library down its left side — *All*, the factory presets, each of your collections, and *Unfiled* for presets saved into none — each with the number of presets in it. Clicking one lists just that one.

IN THE BROWSER	WHAT IT DOES
<code>NEW</code>	Makes a collection. It is listed at once, empty until something goes into it

RENAME	Renames the selected collection; its presets go with it
DELETE	Deletes the selected collection and every preset in it, after asking
Drag a preset onto a collection	Moves it there — or onto <i>Unfiled</i> to take it out of one
Right-click one of your presets	Renames it, changes its description, or deletes it

SAVE asks where a preset goes: pick a collection from the list — or *New collection...* to make one on the spot — then give the preset a name and, if you like, a **description**. The description is what the browser shows while the pointer is over the preset and what the help line shows while it is loaded: a sentence for whoever loads it next. Saving over one of your own presets offers its collection and its description again.

The factory presets are locked. They are part of the plugin, not files of yours, so they cannot be moved, renamed or deleted; to change one, load it, adjust it and save it as your own.

Collections are one level deep on purpose: a preset library is a shelf rather than a filesystem, and a tree deep enough to get lost in is one somebody will get lost in.

PRESET PACKS

A whole collection can be written out as a single file — a **preset pack**, extension `.rainydaypack` — so a library can be handed to somebody else, or moved between machines, as one file rather than a directory.

With a collection selected in the browser:

BUTTON	WHAT IT DOES
EXPORT	Writes the selected collection — its presets and nothing else — as a pack into the <code>packs</code> directory beside <code>presets</code>
EXPORT AS...	The same, to a location you choose
IMPORT...	Lists the packs already in <code>packs</code> , plus <i>Other file...</i> for one from anywhere
REVEAL	Opens the folder the pack was just written to in the system's file browser

Importing adds the pack's presets to the library as a new collection named after the pack. Nothing is ever overwritten: importing the same pack twice gives two collections rather than a mixture of both versions in one.

A pack is the preset format again with a separator line between the presets, so it can be read, diffed and edited by hand like everything else here. It carries each preset's text verbatim rather than a re-serialised copy, which is what makes a round trip through a pack lossless.

EXPORT AS..., *Other file...* and **REVEAL** need the desktop's own file chooser — `zenity` or `kdiallog` on Linux, `xdg-open` to reveal, and the operating system's own on Windows. Where none is installed the buttons that need one are not drawn, and the plugin's own `packs` directory, which is where **EXPORT** writes, is enough on its own.

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/RainyDay/`. `RainyDay_PREFIX=/some/where ./install.sh` overrides the destination. The CLAP headers are found automatically in a few usual places, including a `CLAP/` checkout beside the plugin folder; `-DCLAP_INCLUDE_DIR=/path/to/clap/include` points the build at one kept elsewhere.

A first sound

1. Load RainyDay on an instrument track.
2. Draw a long note — four bars or more. Rain plays 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 **Steady Rain**.

That is the whole workflow. Everything else is shaping.

Three things to try first

Change the surface. *Surface* in the *Rain* panel is the single most audible control in the plugin. The same droplet statistics over Water, Leaves, Metal and Fabric are four different recordings. It biases decay, resonance, the impact click and how far a droplet's pitch bends.

Move it away from you. Raise *Distance*. The rain does not merely get quieter: the air takes the top end off it, the individually audible drops recede and the far-field bed becomes most of what is left. Add *Space Amount* and it is rain heard from inside a building rather than rain in the open.

Sweep the density. *Density* runs from a drip to five thousand drops a second, and it is level-compensated: it changes the texture, not the volume. Sweep it with *Clumping* up and the rain gains surges and lulls instead of an even patter.

How it works

Rain is not one sound. It is a very large number of small independent impacts, plus the collective wash they add up to. RainyDay models exactly that.

Droplet arrival

Droplets are scheduled as a **Poisson point process**: the waiting time until the next impact is drawn as $-\ln(U)/\text{rate}$, the exact inter-arrival distribution for events happening independently at a constant average rate. That is what makes the result sound organic rather than like a machine gun with jitter added.

Clumping turns the rate into a random variable of its own — a *Cox process*, or doubly stochastic Poisson process — modulated by a band-limited Gaussian random walk. The modulation is mean-compensated by $\exp(-\sigma^2/2)$, so the average density stays exactly where it is set while the rain gains natural surges and lulls.

A single droplet

Each impact is a short event assembled from five layers:

LAYER	MODEL	CONTROLLED BY
Bubble	Phase-accumulated sine, a difference of two exponentials for its amplitude, plus a per-droplet pitch sweep	<i>Tonality, Bubble Chance, Chirp, Drop Decay</i>
Second mode	A quieter partial near twice the bubble frequency, decaying twice as fast	<i>Surface</i>
Wet	White-noise burst through a resonant state-variable bandpass tuned to the droplet's pitch	<i>Splash, Tonality</i>
Impact	A two-cycle damped sine at a frequency drawn afresh for every droplet	<i>Impact</i>
Body	One low mode of the struck surface, at the surface's own frequency and ring time	<i>Impact, Surface</i>

Bubble, second mode and splash pass through the droplet's own radiation highpass; the impact and the body do not, because they are the surface being struck rather than the droplet radiating. Everything then passes through a one-pole lowpass standing in for air absorption, and is equal-power panned into the stereo field.

THE SURFACE HAS A VOICE OF ITS OWN

A droplet cannot put energy far below its own resonance, but the thing it lands on can, and every recording of rain on something has more in the low mids than a cloud of droplets radiating into air can produce. The fitted library was short by about 2 dB between 200 and 400 Hz on thirteen presets out of sixteen however each was pointed, which is an engine's bias rather than a preset's.

So each impact also excites one low mode of the surface — 240 Hz and 40 ms for a canopy, 320 Hz and 150 ms for a tin roof, nothing at all for water — scattered a little per droplet, because a roof is not one panel. It is scaled by *Impact*, since it is the strike that sets it going, and weighted as an *energy* ratio against the click rather than an amplitude one, because a mode ringing for forty milliseconds carries far more energy than a two-cycle tick of the same height.

THE IMPACT IS PITCHED, NOT NOISE

Following Liu, Cheng and Tong (2019), the initial impact is modelled as $A \cdot e^{(-2f \cdot t)} \cdot \sin(2\pi f \cdot t)$, with f drawn uniformly between 1 and 16 kHz for each droplet and scaled by the surface's brightness. Damping at twice the frequency leaves about two cycles, so a single drop is a tick with a pitch of its own — 0.2 ms at the top of the range, 3.5 ms at the bottom. One drop sounds like a tick; a thousand a second are broadband, and the constant redrawing is what gives dense rain its shimmer. A fixed noise burst instead gives every drop in the field an identical transient, and measures several dB short of a real recording above 6 kHz.

NOT EVERY DROP RINGS

Pumphrey and Elmore’s measurements have only a band of drop sizes entraining an air bubble on impact; the rest of the rain is splash and tick with no pitch at all. *Bubble Chance* is that fraction, drawn per droplet, and it is not the same control as *Tonality*: tonality at 50 % makes every drop half-pitched, which is a uniform mush, while *Bubble Chance* at 50 % makes half the drops plink clearly and leaves the other half dry.

THERE IS A SECOND BUBBLE MODE

Measuring the isolated drops in the reference recordings finds a partial at 1.8 to 2.15 times the fundamental, 15 to 25 dB below it, on essentially every drop that lands in water: a bubble pulsating hard enough to be heard also radiates at twice its breathing frequency. Its ratio and level are redrawn per droplet, it bends with the fundamental because it is a mode of the same bubble, and it decays twice as fast in dB. Surfaces that trap no bubble do not get it.

THE CHIRP IS REAL PHYSICS

A droplet hitting water entrains an air bubble whose resonant frequency **rises** as the bubble shrinks, which is why a drip into a puddle goes “plink” with an upward bend rather than a flat tone. Two details of that decide whether it is audible.

The bend is large, late, and over before the drop is. Tracked cycle by cycle from the zero crossings of an isolated drop, the reference dips from 775 to 728 Hz across the first four milliseconds, sits on a plateau near 750 Hz while it is within 2 dB of peak, then rises from 846 Hz to 2 kHz between 30 and 110 ms — by which point it is 36 dB down. That is +1.47 octaves in total, and almost none of it happens while the drop is loud. So the model is in two parts: a fast downward dip that relaxes within a cycle or two of the attack, and a rise whose per-sample step grows geometrically. The sweep finishes at 0.6 ring times and then holds, which is the -36 dB point, where the reference has finished too.

The tone arrives behind the splash. The impact happens first and the bubble is entrained afterwards, so the tonal layer’s envelope is $e^{(-t/decay)} - e^{(-t/rise)}$ rather than a decay from full level. That gives it a short swell instead of a hard onset, and how long the swell takes is a property of the surface: water and puddles trap bubbles, metal and glass ring on contact.

Only surfaces that trap a bubble get a large chirp span — Water and Puddle carry 1.30 and 1.45 octaves, while hard surfaces stay near a hundredth of an octave, because a drop landing on something rigid excites a fixed mode of that thing.

Droplet size follows from physics

Rather than randomising amplitude and pitch independently, RainyDay draws a single **size** per droplet from a Marshall–Palmer-like skewed distribution — many small drops, few large ones, with *Level Spread* setting the skew — and derives everything else from it:

- amplitude scales with volume, and so with radius cubed
- resonant pitch scales with $1/radius$: big drops plop low, fine drops tick high
- ring time scales with radius: big drops ring longer

A fat drop is therefore automatically loud, low and long, and a fine one quiet, high and short, with no parameter tweaking required for that to hold. The relations are Minnaert’s, from 1933.

Trickle: drops landing on a surface

A drop landing on concrete makes two sounds, and until 1.7.0 RainyDay modelled only one of them properly. The drop itself is here in detail — the pocket of air it traps, its pitch bend, its splash, the secondary droplets thrown sideways. The *surface* was only a tuned damped sine at the moment of impact, which gives every strike a pitch of its own.

Trickle is the surface, as its own layer, and it is **the same generator RiverFlow uses** for its Trickle — one file shared between the two plugins — so the same settings give the same sound in both. Eight controls, the same eight:

CONTROL	WHAT IT IS
<i>Trickle Level</i>	how much of the layer there is; at minimum it is off
<i>Trickle Rate</i>	drops a second striking the surface; measured 5–25, median 16

<i>Trickle Size</i>	the radius of the pocket of air a drop entrains, in millimetres, which <i>is</i> its pitch
<i>Trickle Spread</i>	the spread of drop sizes, in octaves
<i>Trickle Decay</i>	how long one drop lasts; measured 8–86 ms to fall 10 dB
<i>Trickle Impact</i>	the strike itself against the pocket that follows it
<i>Stone</i>	what they are landing on: the surface's own resonance
<i>Trickle Splash</i>	the wash of water where the surface is deeper

Sizes read in millimetres because that is physically what a pocket's pitch is. Minnaert's relation gives $f_0 \approx 3.26/r$ kHz, and the reference library of running water measures drop pockets at 0.42–1.72 mm with a median of 1.24 — whose predicted 2629 Hz matches a measured median pitch of 2625.

The layer has its own event clock rather than firing with the droplets, which is deliberate: it is a population of drops landing on one nearby hard surface, not the whole field of rain. Setting its rate independently is what lets a sparse patter sit over a busy surface, or the reverse.

The bed

Individually inaudible far-field droplets are not synthesised one by one. They are summed statistically into a **noise bed**: two decorrelated white sources mixed to the requested width — $a \cdot n_1 + b \cdot n_2$ against $a \cdot n_1 - b \cdot n_2$ with $a^2 + b^2 = 1$, so the channel correlation is $\cos(\text{width} \cdot \pi/2)$ with no level change — shaped by a resonant lowpass and a highpass, and modulated by a slow random walk set by *Bed Drift*.

The bed is a band rather than a lowpass, which matters more than it sounds: measured against their own peak, real recordings sit 30 to 50 dB down at 100 Hz, and a lowpass passes everything below its corner flat.

Space and distance

Distance attenuates and darkens, with *Air Absorption* setting how quickly the high end is lost — per droplet, so near drops stay bright while far ones go dull.

Space is a room model driven by one physical quantity. *Space Size* is the dimension of the room, 3 m to 90 m, and everything else follows from it: the distances the first reflections travel, the mean free path the late tank's delay lengths are built on, and — through Sabine's law, with *Space Damping* as the absorption of the surfaces — the decay time. A small room therefore cannot ring for ten seconds and a bare stone hall cannot be dead. The decay is frequency dependent the way real rooms are, so a large space is longer *and* darker.

There are two stages. **Early reflections** are eight discrete taps per channel at fixed fractions of the room dimension; in the reference cave recording the strongest of these arrives 55 ms after each drop only 7 dB below it, and it is the most audible thing about the room. The **late tail** is an eight-line feedback delay network through an orthonormal Hadamard matrix, each line with its own low shelf and air lowpass so that every line decays at the same rate per second whatever its length, with four of the lines slowly modulated by a fraction of a millisecond to break up the metallic modes an unmodulated tank rings with.

Level behaviour

Total loudness is normalised against the *expected number of simultaneously ringing droplets*, which is the rate times the mean ring time. Incoherent sources sum as $\sqrt{N}$, so each droplet is scaled by $1/\sqrt{N}$. That is what makes *Density* a texture control rather than a disguised volume control: sweeping it from a drizzle to a downpour changes the character, not the level. Sparse settings are never scaled *up*, so an isolated drip keeps its natural amplitude. A soft clipper above 0.8 sits at the very end purely as a safety net.

The plugin window

RainyDay 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.

The layout is 960 × 740 in design pixels, and the window resizes by zooming: the host is asked to keep the aspect ratio, and whatever size it settles on becomes one Cairo scale over the same layout, from half size to four times. The panels are the parameter modules and the cells are the parameters, generated from the same table this manual's parameter reference comes from.

The header carries the wordmark, an animated ornament whose density follows how many droplets are sounding, the preset name with its browser, and the activity meter.

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 pan and its width as slim sliders 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.

Saving a preset

SAVE opens a name field and writes the current parameter values to the user preset folder, creating it if it is not there. The name becomes the filename with awkward characters replaced, so *My Rain / 2* is saved as **My_Rain_2.rainyday**, and the browser picks it up immediately.

A name with a slash in it saves into a folder: *Night Walk/Soft Drizzle* writes **Night Walk/Soft_Drizzle.rainyday**, creating the folder if it is not there. See *Folders* above for what the browser then does with it.

Parameter reference

There are 50 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.

OUTPUT

PARAMETER	RANGE	DEFAULT	WHAT IT DOES
Output Gain <code>gain</code>	-inf dB ... 12.00 dB	0.00 dB	Final output level of the whole instrument.

RAIN

PARAMETER	RANGE	DEFAULT	WHAT IT DOES
Density <code>density</code>	0.20 drops/s ... 5.00 kdrops/s	107 drops/s	Average droplet arrival rate. Loudness is compensated, so this is texture, not volume.
Clumping <code>clumping</code>	0.0 % ... 100.0 %	25.0 %	How much the arrival rate fluctuates – surges and lulls instead of an even patter.
Drop Pitch <code>drop_pitch</code>	40.0 Hz ... 9.00 kHz	600 Hz	Base resonant frequency of a droplet. Big drops land below it, fine ones above.
Pitch Spread <code>pitch_spread</code>	0.00 oct ... 5.00 oct	1.60 oct	Random octave spread on top of the pitch that drop size already implies.
Drop Decay <code>drop_decay</code>	1.00 ms ... 1.20 s	46.0 ms	Base ring time of a droplet. The surface scales it further.
Decay Spread <code>decay_spread</code>	0.0 % ... 100.0 %	50.0 %	Randomises ring time from drop to drop.
Tonality <code>tonality</code>	0.0 % ... 100.0 %	35.0 %	Noisy splat at the left, pitched plink at the right.
Impact <code>impact</code>	0.0 % ... 100.0 %	50.0 %	Weight of the broadband click at the moment of impact.
Splash <code>splash</code>	0.0 % ... 100.0 %	35.0 %	Length and weight of the wet noise burst after the impact.
Slosh <code>slosh</code>	0.0 % ... 100.0 %	50.0 %	How long the splash cascades. A drop landing on a hard wet surface – stone, wood or glass – throws secondary droplets that land a few milliseconds apart; this is how many, and so how much it sounds like water rather than a hiss. Surfaces that hold no film of water ignore it.
Level Spread <code>level_spread</code>	0.0 % ... 100.0 %	60.0 %	Skew of the drop-size distribution, which also spreads pitch and decay.
Chirp <code>chirp</code>	-100.0 % ... 100.0 %	35.0 %	Per-droplet pitch bend. Positive rises, the way a bubble in water does.
Surface <code>surface</code>	Water, Puddle, Leaves, Wood, Metal, Glass, Concrete, Fabric	Water	What the rain is falling on. Biases decay, resonance, click and chirp.
Note Tracking <code>note_tracking</code>	0.0 % ... 100.0 %	50.0 %	How far the played MIDI note transposes droplet pitch.
Bubble Chance <code>bubble</code>	0.0 % ... 100.0 %	100.0 %	Fraction of droplets that ring at all. The rest are only splash and click.

DISTANT

PARAMETER	RANGE	DEFAULT	WHAT IT DOES
Bed Level bed_level	-inf dB ... 6.00 dB	-12.00 dB	Level of the far-field wash – the drops too distant to hear individually.
Bed Tone bed_tone	0.0 % ... 100.0 %	50.0 %	Lowpass corner of the bed, dark to bright. Level compensated.
Bed Body bed_body	0.0 % ... 100.0 %	20.0 %	Resonance at the bed’s corner frequency.
Bed Drift bed_drift	0.0 % ... 100.0 %	30.0 %	Slow intensity drift shared by the bed and the droplet rate.
Bed Width bed_width	0.0 % ... 100.0 %	85.0 %	Stereo spread of the far-field bed, independent of the droplets.
Bed Pan bed_pan	-100.0 % ... 100.0 %	0.0 %	Slides the far-field bed left or right without narrowing it.

CLOSE

PARAMETER	RANGE	DEFAULT	WHAT IT DOES
Drop Width width	0.0 % ... 100.0 %	85.0 %	How wide droplets are panned and how decorrelated the bed is.
Drop Pan drop_pan	-100.0 % ... 100.0 %	0.0 %	Slides the close droplets left or right without narrowing them.

SPACE

PARAMETER	RANGE	DEFAULT	WHAT IT DOES
Distance distance	0.0 % ... 100.0 %	30.0 %	Pushes the whole rain field away: quieter, duller, further back.
Air Absorption air	0.0 % ... 100.0 %	50.0 %	How much high end distance costs.
Space Amount space_amount	0.0 % ... 100.0 %	20.0 %	Mix of the room the rain falls in: its first reflections and its tail.
Space Size space_size	0.0 % ... 100.0 %	50.0 %	Dimension of that room, 3 m to 90 m. Reflections and decay time follow from it.
Space Damping space_damping	0.0 % ... 100.0 %	50.0 %	Absorption of its surfaces: bright stone at the left, soft and absorbent at the right.

FILTER

PARAMETER	RANGE	DEFAULT	WHAT IT DOES
Filter Type filter_type	Lowpass, Bandpass, Highpass, Notch	Lowpass	Response of the global output filter.
Filter Cutoff filter_cutoff	20.0 Hz ... 20.00 kHz	20.00 kHz	Corner frequency of the output filter. Fully open at 20 kHz.
Filter Resonance filter_reso	0.0 % ... 100.0 %	10.0 %	Emphasis at the cutoff frequency.
Filter Key Track filter_key_track	0.0 % ... 100.0 %	0.0 %	How far the played note moves the cutoff. Follows the most recent note.
Highpass highpass	20.0 Hz ... 2.00 kHz	20.0 Hz	Rolls off the bottom at 12 dB/oct. Off at the far left.

ENVELOPE

PARAMETER	RANGE	DEFAULT	WHAT IT DOES
Attack attack	1.00 ms ... 15.00 s	200 ms	Fade-in of the rain after a note starts.
Decay decay	1.00 ms ... 15.00 s	599 ms	Fall from full level down to the sustain level.
Sustain sustain	0.0 % ... 100.0 %	100.0 %	Level the rain holds at while the note is held.
Release release	1.00 ms ... 20.00 s	1.20 s	Fade-out after the note is let go.
Velocity to Level vel_to_level	0.0 % ... 100.0 %	50.0 %	How much note velocity controls loudness.
Velocity to Density vel_to_density	0.0 % ... 100.0 %	30.0 %	How much note velocity controls the droplet rate.

SYSTEM

PARAMETER	RANGE	DEFAULT	WHAT IT DOES
Max Droplets max_droplets	32 ... 2048	512	Ceiling on simultaneously sounding droplets. Trade CPU against detail.
Random Seed seed	0 ... 999	0	Starting point of the random sequence, for a repeatable rain.

TRICKLE

PARAMETER	RANGE	DEFAULT	WHAT IT DOES
Trickle Level trickle_level	-inf dB ... 6.00 dB	-30.00 dB	Level of the drops-on-a-surface layer. At minimum it is off and the plugin is droplets and bed alone.
Trickle Rate trickle_rate	0.50 /s ... 200 /s	16.0 /s	Drops a second striking the surface. Measured 5-25 a second across the water references, median 16.
Trickle Size trickle_size	0.15 mm ... 3.00 mm	1.24 mm	Radius of the pocket of air a drop entrains, which is its pitch: Minnaert gives $f_0 = 3.26/r$ kHz. Measured 0.42-1.72 mm, median 1.24 – whose predicted 2629 Hz matches a measured median of 2625.
Trickle Spread trickle_spread	0.00 oct ... 3.00 oct	1.20 oct	Spread of drop sizes. Measured pitches span about 1.25 octaves around the median.
Trickle Decay trickle_decay	2.00 ms ... 200 ms	13.0 ms	How long one drop lasts. Measured 8-86 ms to fall 10 dB, median 13; the long end of that is splashing rather than ticking.
Trickle Impact trickle_impact	0.0 % ... 100.0 %	45.0 %	The strike itself against the pocket that follows it. A drop landing on rock is mostly a broadband tick off the rock; a drop landing in water is mostly the bubble.
Stone stone_tone	800 Hz ... 12.00 kHz	3.50 kHz	What the drops are landing on: the centre of the surface's own resonance. Wet rock and glass are high and hard, wood and moss lower and duller.
Trickle Splash trickle_splash	0.0 % ... 100.0 %	25.0 %	How much of a drop is a short wash of water rather than a clean tick. The references' drop decays are bimodal – 8-13 ms for a tick against 56-86 ms where the water is deeper – and this is that second population.

The preset library

17 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.

Cave Drips

Slow isolated drips falling into a deep stone cavern. Very sparse, each drop short and hard against a long space. *(ambient, texture, cave, sparse, drone)*

Concrete Alley

Rain in a narrow city alley, slapping off hard flat surfaces. *(ambient, texture, urban, reverberant)*

Distant Rain Wall

A heavy downpour heard from far away. All transients lost to air absorption. *(ambient, texture, dark, distant, pad)*

Downpour

Heavy, saturated rainfall. Thousands of impacts per second blurring into a wall of water. *(ambient, texture, noise, loud)*

Dripping Faucet

A single tap that will not stop. Nearly regular timing, tight stereo image, bright and close. *(percussive, sparse, water, foley)*

First Drops

The first scattered drops hitting dry pavement before the real rain arrives. *(ambient, texture, sparse, foley)*

Gutter Trickle

A thin stream running off a downspout. Narrow, wet, and gurgling. *(texture, water, narrow, foley)*

Inside The Car

Muffled rain heard through a windscreen, low-passed and boxy. *(ambient, texture, dark, muffled, lofi)*

Light Drizzle

Fine mist-like drizzle, mostly high-frequency ticking with a soft bright wash. *(ambient, texture, noise, soft)*

Puddle Plinks

Fat drops landing in standing water, each with the slight upward bend of a trapped bubble. *(percussive, sparse, water, playful)*

Rain On Leaves

Forest canopy. Soft, dull, noise-dominated impacts with almost no pitch. *(ambient, texture, organic, soft, dark)*

Steady Rain

Well-behaved everyday rain. The reference point for the whole library. *(ambient, texture, noise)*

Storm Front

A squall line moving through: violent, gusting rainfall that surges and eases. *(ambient, texture, noise, loud, evolving)*

Tin Roof

Rain on corrugated metal. Every impact rings with a bright metallic pitch. *(ambient, texture, metallic, bright)*

Tropical Monsoon

Absurd volumes of warm water falling straight down. Dark, dense and relentless. *(ambient, texture, noise, loud, dark)*

Under an Umbrella

Rain drumming on a taut canopy a foot above your head. Close, blunt and percussive. *(ambient, texture, close, percussive)*

Window Pane

Close-up rain against glass. Hard, bright, glassy ticks right in front of you. *(ambient, texture, bright, glassy, close)*

Using it in a host

Notes

A note starts rain and holding it keeps the rain going; the ADSR shapes the whole field rather than individual droplets, and *Release* is what fades a downpour out. *Note Tracking* decides how far the played note transposes droplet pitch, from none at all — so that any key gives the same rain — to a full transposition, which turns the keyboard into a size control.

Velocity to Level and *Velocity to Density* route velocity, so a soft note can be a lighter shower rather than only a quieter one.

There are 16 voices. Droplets live in one shared pool, so the number of held notes cannot multiply the CPU cost without bound.

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: the engine is one global rain field rather than one per voice, which is also why *Filter Key Track* follows the most recently played note.

Parameter changes are sample accurate, so an automation ramp over a bar is a smooth move and not a staircase at block boundaries.

Reproducible renders

The synthesis is stochastic, so the same project rendered twice does not produce identical audio — which is usually the point, but not always.

Random Seed controls it. At 0 the rain never repeats. At any other value, the same notes render the same rain every time, whatever the plugin was doing before them. Set a non-zero seed before bouncing anything that has to match a previous render, and set it back to 0 afterwards if you want the variation back.

CPU cost

Cost is bounded by design. *Max Droplets* caps the pool, and the level normalisation means a high *Density* spends its budget on more, quieter droplets rather than running away. The far-field bed costs the same whatever its level, because it is two noise sources and two filters however much rain it stands for. *Space* is the most expensive single block: an eight-line feedback delay network plus eight early taps per channel.

If a project is tight on CPU, the order worth cutting in is *Space Amount*, then *Max Droplets*, then *Density*.

Rendering and tails

Rain has a tail: the room, the bed and the last droplets keep sounding after the note is released. When bouncing, leave a few seconds past the end of the note, or the tail is cut off mid-decay.

Reference and limits

Where the model comes from

The droplet model follows the acoustics literature rather than being dialled in by ear:

- Liu, Cheng and Tong, *Physically-based Statistical Simulation of Rain Sound*, ACM TOG 38(4), 2019 — the two-mechanism raindrop model this engine's impact and bubble layers are taken from, and the observation that a bubble is not entrained on every impact.
- Minnaert, *On musical air-bubbles and the sounds of running water*, 1933 — the breathing frequency of a bubble, which is why droplet pitch goes as $1/\text{radius}$ and ring time as radius.
- Marshall and Palmer, *The distribution of raindrops with size*, 1948 — the shape of the drop-size distribution *Level Spread* skews.

Everything the recordings themselves settled — the size of the pitch bend, the ring-time spread, the second bubble mode, how much energy the onset carries above 6 kHz — was measured rather than taken from either.

Notes and limits

- Linux and Windows, x86-64. Plugin state is stored little-endian.
- 16 voices; droplets live in one shared pool.
- Host parameter modulation is supported globally. Per-note modulation and note expressions are ignored.
- *Filter Key Track* follows the most recently played note, because the filter is a single global stage rather than one per voice.
- Deliberately built without `-ffast-math` on x86 GCC that links `crtfastmath.o`, which would flip the FTZ and DAZ flags for the entire host process.
- No wind and no thunder — this is rain only. ThunderClap does the thunder.

Versioning

Semantic versioning. **MAJOR** is an overhaul: a rewrite of the synthesis model, or a change that breaks existing presets or saved host state. **MINOR** adds something — a parameter, a layer, a surface, presets. **PATCH** is fixes that add nothing new.

License

MIT. RainyDay contains no samples and no third-party code. The only external dependencies are the CLAP headers, which are MIT licensed, plus X11 and Cairo for the plugin window.

Reference recordings used during development are not part of the release and are not redistributable.