



VERDALIS
Natural sound synthesis

RiverFlow

Synthesised running water

Version 0.4.0

CLAP instrument for Linux and Windows

Part of the Verdalis Plugin Suite

2026-10-04

Contents

Introduction

- Some rivers really are white noise
- 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 bed, and where its colour comes from
- A river's bands move independently
- Grain, and why it ships at zero
- A dabble is a cluster, not a bubble
- A drop is an impact and then a pocket
- The plunge pool
- Nothing below 60 Hz
- Distance and the banks

The plugin window

- Gestures
- The mixer

Parameter reference

- Flow
- Stones
- Trickle
- Plunge
- Reach
- Filter
- Envelope
- Output

The preset library

Using it in a host

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

What is still being fitted

Reference and limits

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

Introduction

RiverFlow is a CLAP instrument that generates running water: rivers, creeks and waterfalls. It contains no samples. The broadband bed, every pocket of air trapped between the stones and every drop that lands on them are computed from noise, filters and resonators while the plugin plays, so no two takes are alike and there is no loop point to find.

It covers a wide span — from a river loud enough to talk over down to single drops off an overhang — and it covers it that way because the reference library says the two ends are different things rather than the same thing at different volumes.

Some rivers really are white noise

One measurement decides the whole design of this plugin, and it is worth stating plainly because it sounds like a criticism and is not.

Each of the 77 reference recordings was measured for the variation of its 4 ms envelope, band by band, **beside a Gaussian white-noise control put through the same statistics**. The control row is what makes the numbers mean anything: a transient detector finds “events” in band-limited noise at a steady ten to twenty a second, so an event rate quoted for a river without it is not a measurement of the river.

	200-800 HZ	0.8-2 KHZ	2-6 KHZ	6-14 KHZ
Gaussian control	0.30	0.22	0.12	0.09
library minimum	0.30	0.22	0.13	0.11
library median	0.39	0.33	0.30	0.29
library maximum	1.01	0.93	1.64	1.05

The smoothest third of the library lands *on* the control row, and the event detector finds too few discrete events in those recordings to characterise at all. They are not “like” noise. By these measures they are noise, spectrally shaped — and shaped noise is therefore a complete and correct model of them, which is why several of the presets in this manual ship with both event layers switched off.

The other end of the library measures two to thirteen times the control, and the band it happens in says what kind of water it is: around 500 Hz for a creek dabbling between stones, and 2-6 kHz for drops on wet rock, where one reference measures 1.64 against a control of 0.12.

So RiverFlow is a bed plus two populations of discrete events, and the instrument is the ratio between them.

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

RiverFlow 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/RiverFlow/RiverFlow.clap
~/ .clap/RiverFlow/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\RiverFlow\
```

so that `RiverFlow.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*. RiverFlow 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/RiverFlow/presets</code> , or <code>~/ .config/RiverFlow/presets</code>
Windows	<code>%APPDATA%\RiverFlow\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 `.riverflowpack` — 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/RiverFlow/`. `RiverFlow_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 RiverFlow on an instrument track.
2. Draw a long note — eight bars or more. The water runs 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 **Mountain River**, then **Bubbling Creek**, then **Hanging Trickle**.

Those three are the span of the instrument. The first is close to pure shaped noise; the last is almost nothing but countable events.

Three things to try first

Change the water. *Water* selects one of six colours, and they are not names someone invented for a menu: they are the six clusters the reference library falls into, each shipped as the measured octave-band centroid of its members. *Blend* crossfades to the next one, so the six are a continuum rather than six settings. Deep Rush peaks at 500 Hz and falls 25 dB by 16 kHz; Trickle rises all the way to the top.

Turn the events down, then up. Set *Dabble Level* and *Trickle Level* to their minimum and you have a river that is honestly just shaped noise — which is what a third of the reference library measures as. Bring *Dabble Level* up and the water starts folding over stones; bring *Trickle Level* up and drops start landing on them.

Move the water away. Raise *Distance*. Air absorption takes the top off, the individual events multiply and their edges smear, until nothing is countable any more and what is left is the roar. 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 bed, and where its colour comes from

The bed is one noise source per band per channel through eight octave-wide bandpasses from 125 Hz to 16 kHz — which is the resolution the reference colours were measured at.

The gains are not set to the measured curve. They are *solved* so that the bands' **sum** is the measured curve, which is a different thing: octave-wide bandpasses an octave apart overlap by design, and setting each band to its target leaves the two outermost bands 3 to 9 dB low, having a neighbour on one side only. Where a colour's own skirt is steeper than the bank can manage — Creek falls 10 dB from 250 Hz to 125, Deep Rush 10 dB from 8 to 16 kHz — a two-pole filter at the outer band edge makes that skirt and the bank is solved for what is left.

Each band is driven by its **own** noise. That is not an optimisation waiting to be undone: overlapping bandpasses fed one signal add coherently, and solving the gains against a model that assumes their energies add left the steepest colours 7 dB out in their outermost octave. It is also the more faithful arrangement, which is the next section.

A river's bands move independently

The bed's level wanders, and the measurement of *how* is the single thing that most separates this from gated noise.

Two findings. First, the depth of the wander falls with frequency: relative to the 1-2 kHz band, the 125-250 Hz band moves 2.43 times as far and the 4-8 kHz band 0.84 times. That is why a river's low end seems to surge while its hiss sits still, and it is the opposite of what you might guess — more energetic turbulence entrains smaller bubbles, so the top ought to move more. It does not. The reason is probably that the top of the spectrum is the sum of a vast number of tiny bubbles whose average is stable, while the bottom is a handful of large pockets appearing and vanishing in lumps.

Second, the bands do not move **together**. The correlation between one band's 100 ms envelope and its neighbour's measures 0.07 to 0.25 across the library. So the bed is not one gain being moved: each band gets its own random walk, at its own measured depth.

And it is a walk, not an oscillator. The references' envelope spectra fall smoothly at about -1.5 dB per decade over 0.3-10 Hz, with no peak that survives from one recording to the next. A river has no rhythm. *Surge Rate* is therefore the corner of the noise that drives the walk, and there is no LFO anywhere in this plugin.

Grain, and why it ships at zero

Grain modulates each of the bed's eight bands, weighted so that it does far more to the top of the bed than to the bottom. It ships at **zero in every factory preset**, and the story of why is the most instructive thing in this manual.

It was fitted from the 4 ms variance of the library's top band, where the references genuinely do depart from a Gaussian control — 0.29 against 0.09 at 6-14 kHz. So a modulator was built to produce that variance, and it does.

But variance does not say whether a departure is a gentle wobble or a population of discrete events, and the kurtosis of the very same envelopes says which. The references measure an excess kurtosis of 38 in that band. Grain at full produces 7. It matched the number and got the character backwards, and a continuously modulated broadband bed is exactly what *noise* sounds like — which is what a listener reported, in about a minute, by turning this control down and then the bed's level to zero.

The library's own smoothest third settles the physics: those recordings sit *on* the Gaussian control, so a real river's bed really is smooth noise, and whatever graininess a creek has belongs to the events above it rather than to the water underneath them.

The control stays, because it is the cheapest way to thicken a thin bed and a little of it under a sparse preset is useful. It is not a substitute for the event layers, and the presets no longer pretend it is.

A dabble is a cluster, not a bubble

Water folding over a stone traps a pocket of air, and the pocket rings at the Minnaert frequency for its radius: $f_0 \approx 3.26/r$ kHz, so 3.3 mm rings at 988 Hz. The library's dabbles measure a median pitch of 984 Hz. The relation and the recordings agree to one per cent, which is a good reason to trust it.

But one pocket is not what a dabble is, and the measurement is unambiguous about it. The event-triggered spectrum gives a Q between 0.7 and 5, which is a ring of about a millisecond. The event-triggered *envelope* takes 7 to 34 ms to fall 10 dB. One bubble cannot do both. A dabble is a short burst of several — which is also what water folding over a stone physically does — so *Cluster* is how many and *Spill* is how long they are spread over. A cluster shares its pan and its size, because it is one stone.

Glug is how much of a dabble is the water being displaced rather than the air ringing. Without it a cascade of pockets is a music box; with it, it is water.

A drop is an impact and then a pocket

Trickle is single drops striking stone or standing water: the impact off whatever they hit, then the tiny pocket they entrain. Measured 5 to 25 a second at 0.42-1.72 mm, ringing 2-8 kHz — and again Minnaert lands within a per cent of the measured median pitch.

Impact balances the strike against the pocket: a drop on rock is mostly a broadband tick, a drop into water mostly the bubble. *Stone* is what it landed on. *Splash* is the wash where the water is deeper, which exists because the references' drop decays are bimodal — 8-13 ms for a tick against 56-86 ms where there is a pool under it.

The plunge pool

A cloud of bubbles rings far below any bubble in it, which is where the weight under a waterfall comes from. It is not a lowpass of the water; it is a resonance of its own. Three of them, at the 1.00 / 1.53 / 1.90 ratios Xue et al. measure for a pour: one resonator reads as a tuned pipe where three read as a body of water.

Nothing below 60 Hz

Four references carry up to a quarter of their total energy below 60 Hz, and it would be easy to model. By Minnaert, 32 Hz is a 100 mm air pocket — conceivable in the plunge pool under a big fall, out of the question in a creek.

The two candidates are distinguishable. Water in the low band is made by the same events as the water above it, so its envelope correlates with the mid band's; wind and handling noise on the microphone know nothing about the water. Measured, the correlation is below 0.15 in every recording, including the ones where the low end is a fifth of the energy.

It is the recordist's afternoon, not the river. So nothing below 60 Hz is generated, and *Highpass* defaults there for a measured reason rather than a cautious one.

Distance and the banks

Distance is air absorption plus a downward tilt, with the events multiplied and their edges smeared — because a river heard from far off is not one stretch of water heard quietly, it is a whole valley of it arriving over a wide arc.

Banks sets how much is close enough to reflect: 0.03 for an open river against 0.90 for a culvert. This is a lesson the suite learned elsewhere and did not have to learn twice — eight discrete early reflections outdoors is what makes a reverb sound like a bathroom, and no amount of tail talks the ear out of it.

The plugin window

RiverFlow 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 RiverFlow's own river-green theme, with streamlines drifting across the header behind the wordmark and a ring spreading wherever a dabble happens. The rings are deterministic in the dabble 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 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.

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.

FLOW

PARAMETER	RANGE	DEFAULT	WHAT IT DOES
Water <code>water_type</code>	Deep Rush, Rapids, Mountain River, Stream, Creek, Trickle	Stream	Which measured colour the bed has. The six choices are the six clusters the reference library falls into, each shipped as its cluster's measured octave-band centroid – from Deep Rush, which peaks at 500 Hz and falls 25 dB by 16 kHz, to Trickle, which rises all the way to the top.
Blend <code>water_blend</code>	0.0 % ... 100.0 %	0.0 %	Crossfades from the chosen colour towards the next one, so the six measured shapes are a continuum rather than six settings. At 0 the bed is exactly the measured centroid.
Flow Level <code>flow_level</code>	-inf dB ... 6.00 dB	-24.00 dB	Level of the bed, and it reads as exactly that: the bed is normalised to unit RMS before this is applied, whichever colour is selected, so -24 dB is a bed at -24 dBFS and changing colour is not also a level change. The bed is most of a river – the smoothest third of the library is statistically indistinguishable from shaped Gaussian noise, so the bed alone is a complete and correct model of those recordings.
Tilt <code>flow_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>flow_body</code>	0.0 % ... 100.0 %	50.0 %	Weight of the bed's lowest two bands, 125 and 250 Hz. A large river has body here; a creek measures nothing at all below 500 Hz.
Turbulence <code>turbulence</code>	0.0 % ... 100.0 %	22.0 %	How much the bed's level moves, at the 1 kHz band. Measured 5-35% of the mean across the library, median 11%, and the lower bands move 1.5 to 2.4 times further – so a river's low end surges while its hiss sits still. Each band wanders on its own: the measured correlation between one band's envelope and its neighbour's is 0.07-0.25, which is why this cannot be one gain on the whole bed.
Surge Rate <code>surge_rate</code>	0.10 Hz ... 10.0 Hz	1.30 Hz	How fast that movement is. It is a filtered random walk rather than an oscillator: the envelope spectrum of every reference falls smoothly at about -1.5 dB per decade over 0.3-10 Hz with no peak that survives from one recording to the next, so a river has no rhythm. This sets the corner of the noise that drives it.
Grain <code>flow_grain</code>	0.0 % ... 100.0 %	0.0 %	How granular the bed itself is, frequency-weighted so that it does far more to the top of the bed than the bottom. It ships at zero in every factory preset, and the reason is worth knowing before turning it up. It was fitted from the 4 ms variance of the library's top band, where the references do depart from a Gaussian control – but variance does not say whether a departure is a gentle wobble or a population of discrete events, and the kurtosis of the same envelopes says which: the references measure 38 where this measures 7. They are grainy because of events; this is a continuously modulated bed, which is what noise sounds like. The library's smoothest third settles it – those recordings sit <i>on</i> the Gaussian control, so a real river's bed is smooth noise and its grain belongs to the layers above. Kept because it is the cheapest way to thicken a bed, and because a little of it under a sparse preset is useful. It is not a substitute for

the event layers.

STONES

PARAMETER	RANGE	DEFAULT	WHAT IT DOES
Dabble Rate dabble_rate	0.20 /s ... 40.0 /s	5.51 /s	Dabbles per second: water folding over a stone and trapping a pocket of air. Measured 1.4-9.3 a second across the grainy references, median 5.5.
Dabble Level dabble_level	-inf dB ... 6.00 dB	-28.00 dB	Level of the dabbling. Measured 5.5-16.7 dB above the bed it sits on.
Dabble Size dabble_size	1.00 mm ... 12.0 mm	3.30 mm	Radius of a typical trapped pocket, which is its pitch: Minnaert gives $f_0 = 3.26/r$ kHz, so 3.3 mm rings at 988 Hz. The library's measured dabbles are 2.3-5.8 mm, median 3.3 – and their measured median pitch is 984 Hz, which is Minnaert to within a per cent.
Dabble Spread dabble_spread	0.00 oct ... 3.00 oct	0.90 oct	Spread of sizes either side of that. The measured 10th to 90th percentile of dabble pitch spans 562-1406 Hz, which is about two-thirds of an octave each way.
Cluster dabble_cluster	1 ... 12	4	How many pockets one dabble makes. This is measured rather than chosen: the event-triggered spectrum gives a Q of 0.7-5, so a single pocket would ring for about a millisecond, while the event-triggered envelope takes 7-34 ms to fall 10 dB. One bubble cannot do both, so a dabble is a short burst of several – which is also what water folding over a stone does.
Spill dabble_spill	2.00 ms ... 120 ms	25.0 ms	How long a cluster is spread over. Measured 7-34 ms to fall 10 dB, median 25.
Damping dabble_damping	0.25 x ... 4.00 x	1.00 x	Multiplies the damping a pocket physically has. At 1 it is the radiative plus thermal loss after Xue et al.; below 1 rings longer, above 1 is deader.
Glug dabble_glug	0.0 % ... 100.0 %	35.0 %	How much of a dabble is the water being displaced rather than the air ringing. Without it a cascade of pockets is a music box; with it, it is water.

TRICKLE

PARAMETER	RANGE	DEFAULT	WHAT IT DOES
Trickle Rate trickle_rate	0.50 /s ... 200 /s	16.0 /s	Drops a second striking stone or standing water. Measured 5-25 a second, median 16 – the tickling of a small fall is at the top of that.
Trickle Level trickle_level	-inf dB ... 6.00 dB	-24.00 dB	Level of the trickle. Measured 2.2-11.7 dB above the bed: quieter and closer to the water than the dabbling is.
Trickle Size trickle_size	0.15 mm ... 3.00 mm	1.24 mm	Radius of the tiny pocket a drop entrains, again as its pitch. Measured 0.42-1.72 mm, median 1.24, whose Minnaert pitch is 2629 Hz against a measured median of 2625.
Trickle Spread trickle_spread	0.00 oct ... 3.00 oct	1.20 oct	Spread of drop sizes. Measured pitch spans 2.1-5.0 kHz around the median, which is about 1.25 octaves.
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.
Impact trickle_impact	0.0 % ... 100.0 %	45.0 %	The strike itself against the pocket that follows it. A drop landing on rock is a broadband tick; 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 impact's band. Wet rock is high and hard, gravel and moss are lower and duller.
Splash 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 ticks against 56-86 ms where the water is deeper – and this is that second population.

PLUNGE

PARAMETER	RANGE	DEFAULT	WHAT IT DOES
Plunge Level plunge_level	-inf dB ... 6.00 dB	-42.00 dB	The pool under a fall. A cloud of bubbles has collective modes far below any single bubble in it, and that is what the weight under a waterfall is.
Plunge Tone	40.0 Hz ... 600 Hz	180 Hz	The cloud's lowest collective mode. Three are generated, at the 1.00 / 1.53 / 1.90 ratios Xue et al.

plunge_tone	measure for a pour: one resonator is a tuned pipe where three are a body of water.		
Plunge Depth plunge_depth	0.0 % ... 100.0 %	40.0 %	How much the pool surges. A plunge pool is never still.
Plunge Q plunge_q	0.0 % ... 100.0 %	50.0 %	How resonant the pool is. A rock bowl rings; a gravel bed does not.

REACH

PARAMETER	RANGE	DEFAULT	WHAT IT DOES
Banks bank_type	Open, Forest, Gorge, Rock Pool, Culvert, Cavern	Forest	What the water runs between, which is how much early reflected field there is. An open river has nothing close enough to reflect off; a gorge and a culvert have a great deal.
Distance distance	0.0 % ... 100.0 %	30.0 %	How far away the water is. Distance takes the top off it, as air absorption does, and thins the events out.
Air air	0.0 % ... 100.0 %	50.0 %	How much of the high end survives the distance. Cold air over water eats less of it than a warm afternoon does.
Width width	0.0 % ... 100.0 %	55.0 %	Stereo spread of the events. Measured L/R correlation across the library is 0.26-0.99, median 0.60.
Flow Width flow_width	0.0 % ... 100.0 %	30.0 %	Stereo spread of the bed, separately from the events. The bed's L/R correlation is 1 minus this, so the default of 0.40 is the library's measured median correlation of 0.60. A river heard from its bank is wide; one heard down a culvert is not.
Space Amount space_amount	0.0 % ... 100.0 %	22.0 %	How much of the valley, the gorge or the culvert you hear.
Space Size space_size	0.0 % ... 100.0 %	50.0 %	Size of that space, in metres of it.
Space Damping space_damping	0.0 % ... 100.0 %	50.0 %	How absorbent its surfaces are: bare rock rings, a wooded bank does not.

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: four references carry up to a quarter of their energy below 60 Hz, and in every one of them that energy's envelope is uncorrelated with the water above it ($ r < 0.15$). It is wind and handling noise on the microphone, not river, 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 Key Track filter_keytrack	0.0 % ... 100.0 %	0.0 %	How far the filter follows the played note.

ENVELOPE

PARAMETER	RANGE	DEFAULT	WHAT IT DOES
Attack attack	1.00 ms ... 20.00 s	800 ms	Fade-in of the whole scene when a note starts.
Decay decay	5.00 ms ... 20.00 s	499 ms	Fall from the initial level to the sustain.

Sustain sustain	0.0 % ... 100.0 %	100.0 %	Level held while the note is down.
Release release	5.00 ms ... 30.00 s	1.50 s	Fade-out after the note is let go.
Velocity To Level vel_to_level	0.0 % ... 100.0 %	40.0 %	How much velocity sets the level.
Velocity To Flow vel_to_flow	0.0 % ... 100.0 %	30.0 %	How much velocity sets how much water there is: the event rates and the bed's brightness together, which is what more water actually does.

OUTPUT

PARAMETER	RANGE	DEFAULT	WHAT IT DOES
Output Gain gain	-inf dB ... 12.00 dB	0.00 dB	Output level.
Max Events max_events	16 ... 2048	512	Ceiling on pockets and drops alive at once, so the cost is bounded.
Random Seed seed	0 ... 999	0	Non-zero gives the same river every time. Zero is always different.

The preset library

Every preset below is fitted to one named field recording, and checked by rendering it back and measuring it against that recording. Fifteen of the twenty match their reference within 6 dB in every octave band.

Several of them ship with *Grain* at zero and both event layers off. That is not a preset left unfinished — it is what the reference measures as.

20 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 Falls

A big fall into a plunge pool. The weight underneath is the collective modes of the bubble cloud the falling water holds down there, not the bed. (*deep-rush, close*)

Bubbling Creek

The grainiest reference in the library below 800 Hz. This is the dabbling the plugin exists for: water folding over stones, one pocket of air at a time. (*creek, bubbly, tickling, close*)

Bubbly Falls

The bubbliest recording in the library. Almost all of this is countable events; the bed is barely there under them. (*trickle, tickling, close*)

Culvert Drain

Water down a stone culvert. Boxy and close, and everything comes back off the walls. (*stream, tickling, close*)

Distant River

The same water from further off. Nothing is countable any more: distance multiplies the events and smears their edges until they are the roar. (*creek, close*)

Forest Creek

A small creek in woodland, close enough that the individual pockets are countable. (*creek, tickling, close*)

Forest Stream

A stream under trees. The banks are close but soft, so there is a little space and no ring. (*creek, tickling, close*)

Glacier Falls

Meltwater discharging from under a glacier. Hard, cold, and bright at the top. (*creek, tickling, close*)

Gorge Cascade

A cascade down a gorge wall, heard across the gap. Wide and bright, with the rock ringing. (*trickle, close*)

Hanging Trickle

Single drops off an overhang onto wet stone. The far end of the plugin: its 2-6 kHz band varies thirteen times as much as noise would, which is what makes it tick rather than hiss. (*trickle, tickling, close*)

Mountain River

A mountain river in the open, heard from its bank. Almost no low end at all: the reference measures 48 dB down at 32 Hz. (*mountain-river, tickling, close*)

Mountain River Large

The same river with more water in it. Wider and heavier, and the events thin out into the roar. (*mountain-river, close*)

Rapids Close

Standing at the rapids. Loud, close, and all mid band. (*mountain-river, tickling, close*)

River Rush

Water rushing over a shallow bed. Broad and even, with the events well down in it. (*creek, close*)

Small Falls

A small fall over a step of rock into a shallow bowl. *(creek, close)*

Stony Creek

A creek over bare stone. Grainy in every band at once, and bright with it. *(creek, tickling, close)*

Upper Rapids

Rapids in a cut, where the rock is close enough on both sides to send it back. *(rapids, close)*

Wild River

A river running hard through woodland. One of the library's smoothest recordings – statistically it is shaped noise, and this preset is honest about that. *(creek, close)*

Winter River

A river at night in the cold. Cold air absorbs less, so more of the top survives the way over. *(creek, tickling, close)*

Wispy Creek

Barely any water at all. Thin, high and quiet. *(rapids, close)*

Using it in a host

Notes

A note starts the water and holding it keeps it running. The ADSR shapes the whole scene rather than individual events, and *Release* is what fades it out. *Velocity To Flow* makes velocity behave like more water: it raises both event rates and the bed's weight together, which is what a rising river actually does.

Running water needs long notes. Its character is in the statistics of many events and in the slow independent wander of the bed, and neither is audible in a short one.

There are 8 voices, each with its own bed, and 2048 pockets in a shared pool, so the number of held notes cannot multiply the cost without bound. The voice pool is deliberately smaller than elsewhere in the suite because one stretch of water costs sixteen bandpasses a sample.

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 river rising over *Flow Level* and *Dabble Rate* — 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 river 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 river even when note events arrive before the seed has been applied.

CPU cost

Cost is bounded by the pools. The bed is the fixed part — sixteen bandpasses per voice per sample — and it does not vary with any parameter. What varies is *Dabble Rate* times *Cluster*, and *Trickle Rate*, which together decide how many resonators are running, and *Space Amount*, which is the room model. Grain is cheap by construction: it is a multiply per band, not a bank of oscillators.

Rendering and tails

Running water has a short tail compared with surf or thunder — the longest thing still to come after a note is the release plus the room. When bouncing, a second or two past the end of the note is enough.

What is still being fitted

RiverFlow 0.4.0 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 16 kHz octave. Creek and Deep Rush come out about 5 dB high there and within 2 dB everywhere else, and all five presets that miss their reference by more than 6 dB in any band have references peaking in that octave. The cause is understood: an octave-wide bandpass leaks 6-8 dB into its neighbour, so a 15 dB step between the last two bands is out of reach, and one two-pole skirt at the band edge does not close it.

The presets shipped clipping, and now they do not. Ten of the twenty were driven past the output stage's soft clipper — *Hanging Trickle* would have peaked at +12.7 dBFS — which on a noise bed is audible as crackle. Every preset now carries a trim in its own output gain, computed by rendering it 20 dB down and measuring where its peak would land, so all twenty peak at -6 dBFS. The layer levels that carry the fit are untouched. Presets that model quiet water are quiet: a trickle peaking at -6 dBFS sits 19 dB below a waterfall doing the same, which is about the real difference between them.

The most correlated references cannot be reached. Four measure an L/R correlation of 1.00 — one recording of one place — and the renders give 0.77 to 0.88, because a pooled event still lands at one pan and decorrelates whatever it is added to.

The plunge pool, the space controls and *Banks* are chosen, not measured. Separating a plunge pool's resonance from the bed's own low end needs references that isolate one, and the space controls need a reverberation measurement the library cannot give, having neither an impulse nor a known source position.

Reference and limits

Where the model comes from

- Minnaert, M., *On musical air-bubbles and the sounds of running water*, Philosophical Magazine 16, 1933 — $f_0 \approx 3.26/r$, which is why a pocket of air has a pitch at all, and the title of which is this plugin's subject exactly.
- Xue et al., on bubble damping and on the collective modes of a bubble cloud — the radiative and thermal loss terms the pockets use, and the 1.00 / 1.53 / 1.90 mode ratios of the plunge pool.
- Prosperetti, A., on the acoustics of bubble clouds and the origin of the broadband spectrum of running water.
- The reference library itself, which supplied everything else: the six octave-band colours, the per-band surge depths and their independence, the event rates, pitches, ring times and prominences, and the Gaussian control the whole design rests on. The numbers and the scripts that produced them are in the plugin's [tools/analysis/](#).

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 distant weir puts through a floor, that is not in the references and is not in here.
- **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.