SonicBirth — Release Notes

SonicBirth is a modular audio-plugin builder for macOS. Circuits are authored in the design app and exported as Audio Unit, VST, and VST3 plugins.

Version 1.5 — 5 November 2026

Focus: **patching** and **spatial audio**. SonicBirth adopts two patching conventions from Plogue Bidule, and gains an ambisonic encoder up to 7th order.

New — Several wires on one input

- An input can now receive more than one wire. SonicBirth adds the values of all the wires connected to it, so you no longer need an Addition or Add Many element just to mix signals.
- Releasing a new wire on an input that is already connected adds a wire instead of replacing the existing one. To replace a connection, select the old wire and delete it, or drag it away from the input. If several wires are connected to an input, dragging from that input picks up the most recently added wire.
- This applies to number connections, including the outputs of a circuit. Points, audio buffer and FFT sync connections cannot be added together: a new wire still replaces the previous one.
- No extra cost for existing circuits: an input with a single wire works exactly as before. The sum is only calculated for inputs that have two wires or more.
- The same connection cannot be made twice.

New — Connecting several wires at once

- Hold the **Command** key while you drag a wire, either when you click or when you release the mouse. SonicBirth connects the output you started from to the input you release on, then the next output to the next input, and so on, until either element runs out of outputs or inputs.
- It works in both directions (from an output or from an input) and with the inputs and outputs of a circuit. For example, a single Command-drag connects the 64 outputs of an Ambisonic encoder to 64 circuit inputs.
- Command-clicking a connector no longer changes the selection. Command-clicking elsewhere on an element still adds it to or removes it from the selection.

New element — Ambisonic encoder (Miscellaneous)

- Encodes a mono signal into ambisonics, up to 7th order (64 channels).
- Inputs: **in**, **az** (azimuth in degrees, 0 = front, 90 = left), **el** (elevation in degrees, 90 = up), **order** (0 to 7).
- Output format: AmbiX — ACN channel ordering, SN3D normalization, outputs `acn0` to `acn63`.
- Only the channels of the chosen order are calculated: 4 channels at order 1, 16 at order 3, 64 at order 7. The other outputs are silent and cost nothing. The order is read at the start of each audio buffer.
- When the direction does not move during a buffer, the encoding gains are calculated only once for that buffer, which leaves a single multiplication per channel and per sample.

Manual

- The manual covers multiple wires on one input, Command-drag connections and the Ambisonic encoder.

Compatibility

- **Re-export your plugins.** Exported plugins check that the installed SonicBirth framework has the exact version they were exported with. After installing 1.5, plugins exported with an earlier version show *"The framework version does not match the plugin version"* and do not load until they are re-exported from SonicBirth 1.5.
- Circuit files (`.sbc`) from earlier versions open unchanged in 1.5 and sound the same.
- Circuits saved with 1.5 still open in 1.4.x, but an older version keeps only one wire per input and does not know the Ambisonic encoder. Keep using 1.5 for circuits that rely on these features.

Version 1.4.3 — 19 October 2026

Focus: **performance**. Many elements are significantly faster, both in the design app and in exported plugins, with no change to how circuits sound. This release also fixes two bugs in the hosted Audio Unit element.

Faster elements

Figures come from the built-in *Speed test all elements*, comparing 1.4.3 to 1.4.2 on the same Mac. Ranges give the 64-bit and 32-bit precision results.

Element	Speed-up
Points apply	4.5x – 8x
Direction	1.7x – 3.5x
Sine Wave (standard mode)	2.2x – 2.5x
Display Osc. (Var. res.)	2.3x
Lowpass, Highpass, Peak, Notch	1.9x
Bandpass, Bandstop, Crossover	1.9x – 2.1x
BPM Counter	1.5x – 1.6x
Change Slower 2	1.1x – 1.6x
Feedback, Granulate, Granulate with pitch	1.3x – 1.4x
Timer, Cleaner, Formant filter	1.2x – 1.5x

Other improvements that the speed test does not measure: - **Convolving reverb**: 2.6x – 4.1x less CPU, depending on the impulse response length (1 s to 12 s). - **Linear-phase filters**: while a control (frequency, gain, Q, order) is being automated or smoothed, the filter kernel is rebuilt far less often, and each rebuild is 2x – 3x faster. This removes the risk of audio dropouts when automating these filters with small host buffers, especially the sub-bass high-pass. - **Forward FFT / Inverse FFT**: 2x – 3x faster. - **Complex to Polar / Polar to Complex**: 3.5x – 5x faster. - **Denormal protection**: very small numbers produced by decaying filters, feedback loops and reverb tails are now flushed to zero while SonicBirth processes audio. On Intel Macs these numbers can be very slow to compute; on Apple Silicon there is no penalty either way. The host's own setting is restored after each processing call.

All other elements perform as in 1.4.2.

Fixed — AudioUnit element (hosting another Audio Unit)

- In 64-bit (double) precision, the hosted Audio Unit's output could be copied past the end of the buffer when a host buffer was processed in several passes.
- In 64-bit (double) precision, the parameter values sent to the hosted Audio Unit were read incorrectly.

Fixed — Memory

- FFT resources are now released when an FFT element is resized or deleted. Previously they leaked every time.

Notes

- Output is unchanged. The filters (Lowpass, Highpass, Peak, Notch and the elements built from them) produce bit-identical output. Other optimized elements differ from 1.4.2 only by rounding (about 10^{-7} in 32-bit precision).
- The application no longer bundles the obsolete Intel-only `Rez` tool, so it now contains only Universal (Apple Silicon + Intel) code.

Compatibility

- **Re-export your plugins.** Exported plugins check that the installed SonicBirth framework has the exact version they were exported with. After installing 1.4.3, plugins exported with an earlier version show *"The framework version does not match the plugin version"* and do not load until they are re-exported from SonicBirth 1.4.3.
- Circuit files (`.sbc`) are unchanged: circuits from earlier versions open unchanged in 1.4.3, and circuits saved with 1.4.3 can still be opened by 1.4.x.

Version 1.4.2 — 10 August 2026

Focus: a lower-latency **sub-bass linear-phase high-pass**.

Changed — Highpass (lin. phase, sub-bass)

The long-kernel sub-bass high-pass now uses a **65537-tap** kernel instead of 131073 taps. This **halves the latency** to a fixed **65536 samples** (≈1.49 s @ 44.1 kHz, ≈1.37 s @ 48 kHz, down from 131072 samples / ≈2.97 s). In exchange, the lowest usable corner frequency is now **~20 Hz** rather than ~10 Hz — kernel length and the lowest resolvable corner trade off directly, so half the taps doubles the frequency floor.

For the vast majority of rumble and infrasonic cleanup work a 20 Hz floor is more than enough, and the reduced latency makes the element far more practical in a live/monitoring chain. The `f` and `order` controls (order 0 = 12 dB/oct … order 7 = 96 dB/oct) are unchanged.

Action required for existing circuits: if you previously set a circuit's **Latency (samples)** field to **131072** to compensate this element, change it to **65536**. Circuits that referenced the old latency will otherwise be over-compensated.

Notes

- The standard **Highpass (lin. phase)** (4096-sample latency) and all other linear-phase filters are unchanged.
- If you genuinely need a corner below ~20 Hz, the previous behaviour can be restored by building from source with the sub-bass kernel half-length set back to 65536.

Compatibility

- Circuit files (`.sbc`) remain backward compatible: the design app opens any circuit whose saved version is at or below the running version, so 1.4.0 and 1.4.1 circuits open unchanged in 1.4.2.
- Only the sub-bass high-pass latency/frequency floor changed; if a saved circuit relied on the old 131072-sample latency figure, update its **Latency (samples)** field to 65536 (see above).

Version 1.4.1 — 28 July 2026

Focus: a full family of **linear-phase equalizers** plus supporting fixes.

New — Linear-phase filters (Filters category)

Additive companions to the existing (minimum-phase, IIR) filters. Each one matches the **magnitude** response of its IIR counterpart but has **exactly linear phase** (constant group delay, no phase distortion), implemented as a symmetric FIR applied by FFT overlap-add convolution (Accelerate/vDSP), processed internally in double precision.

- **Parametric Eq. (lin. phase)** — inputs `in, f, g, Q` .
- **Peak (lin. phase)** / **Notch (lin. phase)** — inputs `in, f` .
- **Lowpass (lin. phase)** / **Highpass (lin. phase)** — inputs `in, f, order` .
- **Bandpass (lin. phase)** / **Bandstop (lin. phase)** — inputs `in, f1, f2, order` .

All linear-phase filters add a **constant 4096-sample latency** (≈93 ms @ 44.1 kHz), independent of their settings.

New — slope (`order`) control on the slope-type filters

Lowpass, Highpass, Bandpass, and Bandstop take an `order` input that sets the slope in 12 dB/octave steps (order 0 = 12 dB/oct, … order 7 = 96 dB/oct). The whole slope is baked into a single FIR kernel, so **latency stays constant** regardless of the selected order. Bandstop is the exact complement of the bandpass (no crossover phase cancellation).

New — Highpass (lin. phase, sub-bass)

A long-kernel (131073-tap) high-pass for very low corner frequencies **down to ~10 Hz**, where the standard linear-phase high-pass runs out of resolution. Same `f` and `order` controls. Fixed latency of **131072 samples** (≈2.97 s @ 44.1 kHz, ≈2.73 s @ 48 kHz).

Changed

- **Delay (samples)** maximum delay raised from 100000 to **200000 samples**, so a single instance can compensate the sub-bass high-pass latency in one step.

Notes on using linear-phase filters

- Linear phase means a constant processing latency; set the circuit's **Latency (samples)** field to the element's latency so the host aligns the plugin.
- When only some channels/paths are filtered, delay the unfiltered paths by the same amount (e.g. with **Delay (samples)**) to keep everything time-aligned.
- Linear-phase filters exhibit symmetric pre-ring around sharp transients — this is inherent to linear phase, not a defect. For very low, steep corners the minimum-phase IIR remains a good (zero-latency) alternative.

Version 1.4.0 — May 2026

SonicBirth 1.4 adds VST3 plugin export alongside the existing VST2 and Audio Unit formats, introduces four new elements, a performance profiler, and significantly improves audio quality with 4th-order PolyBLEP anti-aliasing and 2x oversampling. The application has also been fully updated to run reliably on current macOS versions and Apple Silicon.

Plugin Export

VST3 Export (New)

- Export your circuits as VST3 plugins, compatible with all major modern DAWs
- All knobs, sliders, switches, and indexed lists are exposed as automatable parameters
- MIDI input: note-on, note-off, and poly pressure
- Tempo and time position sync when your circuit uses them
- Circuits with side chain inputs create the appropriate auxiliary bus in the host
- Correct latency and tail time reporting
- Single, install, and batch export from the File menu (Ctrl+Cmd shortcuts), same as VST2
- SonicBirth plugins can now be used in DAWs that have dropped VST2 support

VST2 Export: GUI Fixed

- Fixed a bug where exported VST2 plugins were not displaying their custom interface in some hosts

Audio Unit Export Updated

- Audio Unit plugins are now compatible with current macOS versions

Rendering Engine Updated

SonicBirth's display system has been updated to use Apple's native graphics framework, replacing OpenGL which Apple deprecated in macOS 10.14. All elements, displays, and circuit views look and behave exactly the same, but the update brings full Retina display support and reliable rendering on both Intel and Apple Silicon Macs.

Audio Engine Updated

The audio engine has been updated to use Apple's modern audio framework, replacing an unmaintained third-party library. This improves stability and compatibility with current macOS and Apple Silicon hardware. There is no change to how circuits sound or behave.

New Elements

Triggered Random (`trnd`)

- **Category:** Generator
- **Inputs:** `t` (trigger — fires on rising edges crossing 0.5), `type` (mode), `range` (step size, 0–1), `bias` (–1 to +1), `states` (2–25)
- **Outputs:** `out` (generated value), `state` (Markov state index, 0 otherwise)
- On each rising edge of `t`, generates a new value according to the selected mode; output holds until the next trigger
- **Mode 0 — Coin:** outputs 0 or 1 with equal probability. Useful for random gates, switches, and stochastic music
- **Mode 1 — Random:** picks any value in [–1, 1]. Triggered equivalent of the Random element
- **Mode 2 — Drunk:** bounded random walk in [–1, 1] with reflection; `range` scales the maximum step size
- **Mode 3 — Markov:** transitions between evenly-spaced states; `range` scales the maximum jump distance, `bias` shifts jumps toward lower (<0) or higher (>0) states
- `type`, `range`, `bias`, and `states` are read once per audio buffer

Feedback Zero (`fdbck0`)

- **Category:** Feedback
- **Purpose:** Feedback with only 1 sample of delay (~0.02 ms at 44.1 kHz)
- Use when tight feedback loops require less latency than the standard Feedback element's 10 ms delay — for example in self-oscillating filters or oscillators built from feedback
- **Note:** Both Feedback and Feedback Zero introduce an additional, intrinsic delay equal to the host's audio buffer size. This is unavoidable and inherent to how audio buffering works.

Change Slower 2 (`slow2`)

- **Category:** Misc
- **Inputs:** `i` (value to smooth), `time` (transition duration in ms), `type` (easing curve, 0–16)
- Like the original Change Slower, but with 17 selectable easing curves:
  - 0 = Sine (default)
  - 1 = Smootherstep
  - 2 = Smoothstep
  - 3 = Cubic
  - 4 = Quadratic
  - 5 = Quartic
  - 6 = Quintic
  - 7 = Circular
  - 8 = Linear
  - 9 = Elastic + Cubic
  - 10 = Elastic + Smoothstep
  - 11 = Elastic + Smootherstep
  - 12 = In (accelerate)
  - 13 = Out (decelerate)
  - 14 = Fast Start
  - 15 = Slow Start
  - 16 = Exponential (like the original Change Slower)

Moving Average (`mavg`)

- **Category:** Misc
- **Inputs:** `i` (signal), `time` (window size in ms)
- **Output:** `o` (averaged signal)
- Smooths a signal by averaging it over a sliding time window up to 10 seconds long
- Window size can be changed dynamically

Updated Elements

Sine Wave (`sine`)

- New `type` input selects between standard sine (0, default) and the Bhaskara I fast approximation (1)
- The fast mode is approximately 2× cheaper in CPU, with a maximum error of 0.16% (−56 dB) — inaudible in practice
- Existing patches using Sine Wave are unaffected; the standard mode remains the default

Random (`rnd`) and Random Ramp (`rndr`)

- New `type` input selects the generation mode:
    - 0 = Standard random (default — existing behavior)
    - 1 = Drunk walk — takes a random step of up to `range` at each trigger, bouncing off the ±1 boundaries. Produces a bounded wandering signal, similar to Max/MSP's drunk object
    - 2 = Markov — jumps between `states` evenly-spaced positions across ±1, with jumps scaled by `range` and a directional `bias` (-1 toward lower states, +1 toward higher). A second output `state` gives the current state index
- New `range` input (0–1): maximum step size for drunk walk and Markov modes
- New `bias` input (−1 to +1): directional tendency in Markov mode
- New `states` input (2–25): number of discrete states in Markov mode
- Random Ramp smoothly interpolates between values in all three modes

White Noise, Pink Noise, Random, Random Ramp, Linear Noise, Triggered Random

- The random number generator has been upgraded to a higher-quality algorithm (xoshiro256++) across all of these elements
- The previous generator had a relatively short period and statistical weaknesses audible as subtle patterns or coloration in noise. The new one has a period of $2^{256} - 1$ and passes all standard randomness test suites — white noise sounds cleaner and more neutral, and generative sequences are less predictable over long durations
- Each element instance now has its own independent random state. Previously, multiple noise or random elements in the same circuit could produce correlated output; they now evolve fully independently

Elements Profiler (New)

A new floating panel for measuring the CPU usage of individual elements in a circuit.

- Shows each element's name, percentage of total CPU time, time per call in microseconds, and call count
- Sub-circuits are expandable to see a per-element breakdown inside them
- Selecting an element in the profiler highlights it in the circuit view, and vice versa
- **A/B Snapshot Comparison:** Capture two snapshots ("Snap A" and "Snap B") and compare them side by side with a delta column sorted by the largest change — useful for measuring the effect of circuit optimizations
- Sound playback starts automatically when profiling begins

Equation Enhancements

Comparison Functions (return 1.0 or 0.0)

- `lt(a,b)` — less than
- `le(a,b)` — less than or equal

- `gt(a,b)` — greater than
- `ge(a,b)` — greater than or equal
- `eq(a,b)` — equal
- `ne(a,b)` — not equal

Logical Functions (return 1.0 or 0.0)

- `and(a,b)` — both non-zero
- `or(a,b)` — either non-zero
- `not(a)` — zero becomes 1, non-zero becomes 0

Utility Functions

- `sign(x)` — returns –1, 0, or 1
- `clamp(x, lo, hi)` — clamps value to [lo, hi]

Conditional Select Functions

- `iflt(a, b, c, d)` — if a < b then c, else d
- `ifle(a, b, c, d)` — if a ≤ b then c, else d
- `ifgt(a, b, c, d)` — if a > b then c, else d
- `ifge(a, b, c, d)` — if a ≥ b then c, else d
- `ifeq(a, b, c, d)` — if a == b then c, else d
- `ifne(a, b, c, d)` — if a ≠ b then c, else d
- Enable signal-dependent routing directly in equations, e.g. `ifgt(i0, 0.5, i1, i2)` selects between two inputs based on a control signal
- Inline constants are accepted in any argument position: `ifeq(i0, 1, i1, 0)` and `clamp(i0, -1, 1)` are valid. **Note:** the first argument (`a`) of a conditional select must be a signal input, not an inline constant

Performance: Equations with Inline Constants

- Inline constants in equations (such as the `1` and `0` in `ifeq(i0, 1, i1, 0)`) are pre-computed once when the circuit is set up, carrying no processing cost at audio rate
- Equations mixing signals and constants (conditional selects, `clamp` , arithmetic with literal values) will show a measurable CPU reduction, visible in the Elements Profiler

Performance: Apple Silicon

- All comparison, logical, and sign operations use NEON SIMD on Apple Silicon, processing four samples at a time
- Significant CPU reduction for equation-heavy circuits on Apple Silicon Macs

Audio Quality Improvements

Anti-Aliasing: 4th-Order PolyBLEP / PolyBLAMP

Saw, square, and triangle oscillators now use advanced bandlimited anti-aliasing: - **PolyBLEP** corrects step discontinuities (saw and square edges) - **PolyBLAMP** corrects slope discontinuities (triangle peaks and troughs) - 4th-order polynomials provide approximately 12 dB more aliasing reduction than conventional 2nd-order PolyBLEP

2x Oversampling

All non-sinusoidal oscillators (saw, square, triangle) now run internally at twice the sample rate, then are filtered and decimated back to the output rate. This greatly reduces aliasing, especially at lower sample rates, at a modest CPU cost.

The combination of 4th-order PolyBLEP and 2x oversampling produces clean, professional-quality waveforms. The improvement is most audible on saw and square waves at high pitches or low sample rates.

Faster Sine Wave

The Bhaskara I approximation added to the Sine Wave element is approximately 2× faster than a standard sine calculation, with a maximum error of 0.16% (–56 dB) — inaudible in practice.

UI Improvements

Enhanced Field Borders

Editable text fields and tables throughout the application now have consistent, visible borders, improving legibility in settings and circuit views.

Manual Improvements

- All function signatures and input/output names in the Equation section use monospace font, matching how they appear in the element's text field
- Monospace font size adjusted to match the x-height of the surrounding body text
- Limitation note added to the Conditional Select section clarifying that the first argument must be a signal

Resizable Panels

- The Info and Settings panels can now be resized by dragging
- The Settings panel automatically adjusts its height to fit the current element's settings

System Requirements

- **macOS 12.4** (Monterey) or later
- **Universal Binary** — runs natively on both Intel and Apple Silicon Macs

Backward Compatibility

- All existing SonicBirth documents (.sbc files) open without changes
- The original Change Slower element is preserved; Change Slower 2 is a separate element
- The Sine Wave element's new fast mode is opt-in (type = 1); all existing patches default to standard mode
- VST2 and Audio Unit export continue to work alongside the new VST3 export

Summary of Changes

Category	Change
Plugin Export	VST3 export (new), VST2 GUI fix, AU compatibility update
Rendering	Updated to Apple's Core Graphics — Retina support, no deprecated APIs
Audio Engine	Updated to Apple's AVAudioEngine — better stability and Apple Silicon support
New Elements	Triggered Random, Feedback Zero, Change Slower 2, Moving Average
Updated Elements	Sine Wave (fast mode), Random + Random Ramp (drunk walk, Markov chain modes), all random elements (improved PRNG)
Profiler	New Elements Profiler with A/B snapshot comparison
Equation	Comparisons, logic, sign, clamp, conditional selects, mixed signal/constant args, constant pre-computation, Apple Silicon SIMD
Audio Quality	4th-order PolyBLEP/BLAMP, 2x oversampling, faster sine option
UI	Field borders, resizable panels, manual improvements
Requirements	macOS 12.4+, Universal Binary (Intel + Apple Silicon)

Compatibility

- Circuit files (`.sbc`) are backward compatible: the design app opens any circuit whose saved version is at or below the running version, so 1.4.0 circuits open unchanged in 1.4.1.
- The new elements are purely additive — no existing element behavior was changed.

SonicBirth 1.5 — 5 November 2026. Originally created by Antoine Missout. Continued by Daniel Courville.