

SonicBirth 1.4 Release Notes

Overview

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 2x 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 $a \leq b$ then c , else d
- `ifgt(a, b, c, d)` — if $a > b$ then c , else d
- `ifge(a, b, c, d)` — if $a \geq b$ then c , else d
- `ifeq(a, b, c, d)` — if $a == b$ then c , else d
- `ifne(a, b, c, d)` — if $a \neq b$ 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 2x 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) |

SonicBirth 1.4 — May 2026. Originally created by Antoine Missout. Continued by Daniel Courville.