Container Port Specification v1.2

Overview

This document specifies the inputs, outputs, and configuration for each container type in the Calamus sound engine. These containers form the building blocks of sounits — connected via the tensor architecture to create sound bodies.

Design Principles

No Hard-Coded Limits

All numeric limits are configurable via Preferences. Reasonable defaults are provided, but the user can extend any range. This enables unexpected discoveries — like the 1,400 harmonics that created standing waves in the room.

Implementation: A Limits/Constraints class provides min/max values. Containers query this class rather than using magic numbers.

Everything Modulatable is an Input

Any parameter that could benefit from modulation is exposed as an Input port with a default value. Leave it unconnected — uses default. Plug something in — it moves.

Result: Maximum flexibility. An envelope, easing curve, physics system, or any control source can modulate any parameter.

Modular Composition

Containers can be combined freely:

- Instantiate any container multiple times (3 HarmonicGenerators? Sure.)
- Chain containers in any order (with type compatibility)
- The signal path MUST end with a Signal output
- Skip any container by simply not including it

Container Categories

Essential: Required for any sound

- HarmonicGenerator — creates the spectrum
- SpectrumToSignal — converts to audio (must have one to produce sound)

Shaping: Transform the sound character

- RolloffProcessor — brightness control
- FormantBody — vowel/resonance (add more instances for F3, F4, etc.)
- BreathTurbulence — voice/noise blend
- NoiseColorFilter — noise spectrum shaping

Modifiers: Affect how parameters change over time

- PhysicsSystem — mass/spring/damping (organic movement + click prevention)
- EasingApplicator — shaped transitions between values
- EnvelopeEngine — parameter evolution over note lifetime
- DriftEngine — micro-detuning for organic quality
- GateProcessor — note on/off lifecycle

Data Types

Signal — Audio-rate stream of samples (float, -1.0 to 1.0)

Spectrum — Array of harmonic amplitudes indexed by harmonic number

Control — Single value, can change at control rate or audio rate

Trigger — Discrete event (gate on, gate off, impulse)

Connection Functions

When connecting an output to an input, a function determines how values combine:

- passthrough — source replaces input entirely (ignores weight)
- add — output added to input (input + source × weight)
- multiply — output multiplied with input (input × source × weight)
- subtract — output subtracted from input (input - source × weight)
- replace — weighted blend (input × (1-weight) + source × weight)
- modulate — bipolar modulation around input (input + (source - 0.5) × weight × range)

Envelope vs Easing

Both modify parameters over time, but serve different purposes:

Envelope (EnvelopeEngine):

- Arbitrary curve shape (drawn, preset, mathematical)
- Spans the note lifetime (or loops)
- Defines WHAT happens over time
- Example: 'Breath swells during note release'

Easing (EasingApplicator):

- Mathematical function (elastic, bounce, cubic, etc.)
- Spans a transition between two values
- Defines HOW movement happens
- Example: 'Pitch glides from A to B with elastic bounce'

They work together: Easing shapes the character of movement; Envelope defines the journey.

Reference Signal Flow

This is the typical 'human voice' configuration. The tensor architecture allows any valid connection.

Core signal path:

- Pitch → HarmonicGenerator → RolloffProcessor → SpectrumToSignal →
- → BreathTurbulence (+ NoiseColorFilter noise input) →
- → FormantBody → Output Signal

Modifier connections:

- PhysicsSystem → FormantBody.f1Freq, FormantBody.f2Freq (smooth movement)
- PhysicsSystem → RolloffProcessor.rolloff (brightness bloom)
- DriftEngine → HarmonicGenerator.drift (organic beating)
- GateProcessor.attackTrigger → PhysicsSystem.impulse (consonant kick)
- EnvelopeEngine → BreathTurbulence.blend (breath swell on release)

1. HarmonicGenerator

Generates the fundamental harmonic spectrum — the 'bones' of the sound.

Ports

Port Name	Direction	Data Type	Default	Description
pitch	Input	Control	—	Fundamental frequency in Hz (required)
numHarmonics	Input	Control	64	Number of harmonics to generate
dnaSelect	Input	Control	0	DNA sequence index (preset or custom)
purity	Input	Control	1.0	Blend: 0 = pure DNA, 1 = all harmonics
drift	Input	Control	0.0	Per-harmonic detuning (from DriftEngine)
spectrum	Output	Spectrum	—	Array of harmonic amplitudes

Configuration (Preferences)

Config Option	Type	Default	Description
maxHarmonics	int	64	Upper limit for numHarmonics (try 1400 for standing waves!)
dnaPresets	list	6 presets	All, Odd, Even, OddDominant, Fundamental+, Octaves
phaseMode	enum	random	Initial phase: zero, random, custom

Processing

For each harmonic h (1 to numHarmonics): amplitude[h] = dnaWeight(h) × purity + (1-purity). Output spectrum contains amplitude per harmonic. Phase accumulators maintained internally.

2. RolloffProcessor

Applies brightness curve to spectrum — the 'magic ingredient' that prevents static sound.

Ports

Port Name	Direction	Data Type	Default	Description
spectrumIn	Input	Spectrum	—	Harmonic amplitudes from HarmonicGenerator
rolloff	Input	Control	0.6	Rolloff power (higher = darker)
spectrumOut	Output	Spectrum	—	Modified harmonic amplitudes

Configuration (Preferences)

Config Option	Type	Default	Description
rolloffMin	float	0.1	Minimum rolloff value
rolloffMax	float	3.0	Maximum rolloff value
rolloffCurve	enum	power	Shape: power, exponential, custom

Processing

For each harmonic h: amplitudeOut[h] = amplitudeIn[h] / pow(h, rolloff). Critical discovery: this is what makes 64 harmonics sound different from 4 harmonics.

3. SpectrumToSignal

Converts spectrum data to audio signal by summing oscillators. Essential — must have one to produce sound.

Ports

Port Name	Direction	Data Type	Default	Description
spectrumIn	Input	Spectrum	—	Harmonic amplitudes
pitch	Input	Control	—	Fundamental frequency for oscillators
normalize	Input	Control	1.0	Auto-normalize: 0 = off, 1 = full
signalOut	Output	Signal	—	Summed audio signal

Configuration (Preferences)

Config Option	Type	Default	Description
antiAlias	bool	true	Skip harmonics above Nyquist

Processing

Maintains phase accumulator per harmonic. Each sample: sum all sin(phase[h]) × amplitude[h], advance phases. Normalization prevents clipping when many harmonics sum.

4. FormantBody

Resonant filtering creating vowel-like character — the 'throat'. Add multiple instances for F3, F4, etc.

Ports

Port Name	Direction	Data Type	Default	Description
signalIn	Input	Signal	—	Audio to be filtered
f1Freq	Input	Control	500	First formant frequency (Hz)
f2Freq	Input	Control	1500	Second formant frequency (Hz)
f1Q	Input	Control	8.0	First formant resonance
f2Q	Input	Control	10.0	Second formant resonance
directMix	Input	Control	0.3	Dry/wet: 0 = all filtered, 1 = all direct
f1f2Balance	Input	Control	0.6	Balance: 0 = F2 only, 1 = F1 only
signalOut	Output	Signal	—	Filtered audio

Configuration (Preferences)

Config Option	Type	Default	Description
f1FreqMin	float	200	F1 minimum Hz
f1FreqMax	float	1000	F1 maximum Hz
f2FreqMin	float	500	F2 minimum Hz
f2FreqMax	float	3000	F2 maximum Hz
qMin	float	1.0	Minimum Q factor
qMax	float	20.0	Maximum Q factor

Processing

Two parallel resonant bandpass filters (biquad). Output = directMix × input + (1-directMix) × (F1 × f1f2Balance + F2 × (1-f1f2Balance)).

Click prevention: Route f1Freq and f2Freq through PhysicsSystem before connecting. The smoothing prevents coefficient discontinuities that cause audio glitches.

5. BreathTurbulence

Blends harmonic content with noise — adds the 'breath' that makes sound human.

Ports

Port Name	Direction	Data Type	Default	Description
voiceIn	Input	Signal	—	Harmonic/voiced signal
noiseIn	Input	Signal	—	Noise signal (from NoiseColorFilter)
blend	Input	Control	0.10	Mix: 0 = pure voice, 1 = pure noise
signalOut	Output	Signal	—	Blended output

Configuration (Preferences)

Config Option	Type	Default	Description
blendCurve	enum	sqrt	Mapping: linear, sqrt, log

Processing

Output = voiceIn × (1 - blend) + noiseIn × blend.

Critical discovery: Just 5-15% noise transforms electronic sound into something with lungs, throat, and body.

6. NoiseColorFilter

Shapes noise spectrum — from dark breath to bright fricatives.

Ports

Port Name	Direction	Data Type	Default	Description
noiseIn	Input	Signal	(internal)	External noise or use internal generator
color	Input	Control	2000	Filter cutoff frequency (Hz)
filterQ	Input	Control	1.0	Filter resonance
noiseOut	Output	Signal	—	Colored noise

Configuration (Preferences)

Config Option	Type	Default	Description
useInternal	bool	true	Generate white noise internally
filterType	enum	highpass	Filter: lowpass, highpass, bandpass
colorMin	float	100	Minimum cutoff Hz
colorMax	float	12000	Maximum cutoff Hz

Processing

Low color (100 Hz) = dark, breathy rumble. Mid (2000 Hz) = 'SH' territory. High (8000 Hz) = 'S' territory, hissy.

7. DriftEngine

Adds micro-detuning to harmonics — subtle beating that creates organic quality.

Ports

Port Name	Direction	Data Type	Default	Description
amount	Input	Control	0.005	Drift intensity (0 = none)
rate	Input	Control	0.5	How quickly drift wanders (Hz)
driftOut	Output	Control	—	Per-harmonic detuning multipliers

Configuration (Preferences)

Config Option	Type	Default	Description
amountMax	float	0.1	Maximum drift (10%)
rateMax	float	10.0	Maximum drift rate Hz
driftPattern	enum	perlin	Pattern: random, sine, perlin
perHarmonic	bool	true	Each harmonic drifts independently

Processing

Slow-moving LFOs per harmonic. Output: array of multipliers near 1.0. Creates subtle beating between partials — the sound 'breathes' even on sustained notes.

8. PhysicsSystem

Applies mass/spring/damping to parameter changes — the core 'organic without random' mechanism.

Ports

Port Name	Direction	Data Type	Default	Description
targetValue	Input	Control	—	Where the parameter wants to go
mass	Input	Control	0.5	Resistance to change (0 = instant)
springK	Input	Control	0.001	Spring constant (pull toward target)
damping	Input	Control	0.995	Energy loss per sample
impulse	Input	Trigger	—	Velocity kick (for attacks)
impulseAmount	Input	Control	100	Strength of impulse
currentValue	Output	Control	—	Smoothed value with physics

Configuration (Preferences)

Config Option	Type	Default	Description
massMax	float	1.0	Maximum mass
velocityLimit	float	1000	Clamp to prevent instability

Processing

Per sample: force = (target - current) × spring; velocity += force; velocity *= damping; current += velocity. On impulse: velocity += impulseAmount.

Dual purpose: Expression (organic movement) AND stability (prevents clicks from abrupt parameter changes).

9. EasingApplicator

Shapes parameter transitions using easing functions — prescribed rather than emergent movement.

Ports

Port Name	Direction	Data Type	Default	Description
startValue	Input	Control	0.0	Beginning of transition
endValue	Input	Control	1.0	End of transition
progress	Input	Control	—	Position in transition (0-1)
easingSelect	Input	Control	0	Which easing function
easedValue	Output	Control	—	Result after easing

Configuration (Preferences)

Config Option	Type	Default	Description
easingLibrary	ref	built-in	Available easing functions
defaultEasing	enum	linear	Default when easingSelect not connected
defaultMode	enum	inOut	Default: in, out, inOut

Processing

t_eased = easingFunction(progress); output = startValue + (endValue - startValue) × t_eased.

Use cases: Pitch glide between notes, formant sweeps for consonants, brightness bloom on attack.

10. GateProcessor

Handles note on/off lifecycle — the birth and death of sound.

Ports

Port Name	Direction	Data Type	Default	Description
			—	Assumed active for lifetime of the note
velocity	Input	Control	1.0	Note velocity
attackTime	Input	Control	0.01	Attack duration (seconds)
releaseTime	Input	Control	0.1	Release duration (seconds)
attackCurve	Input	Control	0	Easing function for attack
releaseCurve	Input	Control	0	Easing function for release
velocitySens	Input	Control	0.5	How much velocity affects output
envelopeOut	Output	Control	—	Current envelope value (0-1)
stateOut	Output	Control	—	State: 0=off, 1=attack, 2=sustain, 3=release
attackTrigger	Output	Trigger	—	Fires once at note start
releaseTrigger	Output	Trigger	—	Fires once at note end

Configuration (Preferences)

Config Option	Type	Default	Description
attackTimeMin	float	0.001	Minimum attack (1ms)
attackTimeMax	float	5.0	Maximum attack
releaseTimeMin	float	0.01	Minimum release
releaseTimeMax	float	10.0	Maximum release

Processing

State machine: OFF → ATTACK → SUSTAIN → RELEASE → OFF. EnvelopeOut follows attack/release curves. AttackTrigger routes to PhysicsSystem.impulse for 'Ba Ba' consonant effect.

11. EnvelopeEngine

Applies arbitrary envelope shapes to parameters over note lifetime.

Ports

Port Name	Direction	Data Type	Default	Description
			—	Assumed active for lifetime of the note
noteProgress	Input	Control	—	Position in note (0-1)
envelopeSelect	Input	Control	0	Which envelope curve
timeScale	Input	Control	1.0	Stretch/compress duration
valueScale	Input	Control	1.0	Multiply output
valueOffset	Input	Control	0.0	Add to output
envelopeValue	Output	Control	—	Current envelope value

Configuration (Preferences)

Config Option	Type	Default	Description
envelopeLibrary	ref	built-in	Available envelope curves
loopMode	enum	none	Behavior at end: none, loop, pingpong

Processing

On trigger: reset position. Each sample: read envelope at position × timeScale, apply valueScale and valueOffset. Multiple instances can run in parallel for different parameters.

Use case: Ben Webster's breathy release — breath blend increases as note fades. Brightness that evolves over sustained notes. Vibrato that develops over time.

Typical Sounit Configurations

Minimal (Essential Only)

HarmonicGenerator → RolloffProcessor → SpectrumToSignal → [Output]

Just harmonics and brightness. No breath, no formants, no physics.

Voice Character

HarmonicGenerator → RolloffProcessor → SpectrumToSignal → BreathTurbulence → FormantBody → [Output]

With: NoiseColorFilter → BreathTurbulence.noiseIn

Adds throat resonance and breath. Sounds human-ish.

'Ba Ba' Configuration

Voice Character setup, plus:

- PhysicsSystem(A) → FormantBody.f1Freq
- PhysicsSystem(B) → FormantBody.f2Freq
- GateProcessor.attackTrigger → PhysicsSystem(A).impulse
- GateProcessor.attackTrigger → PhysicsSystem(B).impulse

The consonant 'B' emerges from formant trajectories kicked by attack impulse.

Full Expression

All containers, with:

- PhysicsSystem on every modulatable frequency/level parameter
- DriftEngine → HarmonicGenerator.drift
- EasingApplicator on pitch for glides
- EnvelopeEngine on breath blend for release swell
- Multiple FormantBody instances for F1, F2, F3, F4

Summary

11 Containers:

- 2 Essential: HarmonicGenerator, SpectrumToSignal
- 4 Shaping: RolloffProcessor, FormantBody, BreathTurbulence, NoiseColorFilter
- 5 Modifiers: PhysicsSystem, EasingApplicator, EnvelopeEngine, DriftEngine, GateProcessor

Key Principles:

- No hard-coded limits — all configurable via Preferences
- Everything modulatable is an input with a default
- Multiple instances of any container allowed
- Must end with Signal output
- PhysicsSystem provides both expression AND stability

Version 1.1 — December 2025