

# Ocean Swift - Sound Creator Suite

## User Manual

### Table Of Contents

|                                           |           |
|-------------------------------------------|-----------|
| <b>Table Of Contents.....</b>             | <b>1</b>  |
| <b>Introduction.....</b>                  | <b>9</b>  |
| <b>Navigation and Tool Selection.....</b> | <b>10</b> |
| <b>Settings and Configuration.....</b>    | <b>10</b> |
| Audio and MIDI Settings.....              | 10        |
| Kontakt Developer Integration.....        | 10        |
| Application Preferences.....              | 11        |
| <b>The Log Viewer.....</b>                | <b>11</b> |
| Log Controls.....                         | 11        |
| Log Filtering.....                        | 11        |
| <b>Safe Shutdown Policy.....</b>          | <b>12</b> |
| <b>The Tools.....</b>                     | <b>12</b> |
| Overview.....                             | 12        |
| Audio Batch Process.....                  | 14        |
| Introduction.....                         | 14        |
| Global Workflow & Presets.....            | 14        |
| Spectral & Pitch Processing.....          | 15        |
| Phase & Utility.....                      | 15        |
| Time & Amplitude Envelopes.....           | 15        |
| Saturation & Analog Character.....        | 16        |
| Leveling & Output Formatting.....         | 16        |
| Loop Detection.....                       | 17        |
| Internal Signal Flow Order.....           | 17        |
| Audio Loop Aligner.....                   | 17        |
| Introduction.....                         | 17        |
| Universal Format Engine.....              | 18        |
| Interface Guide.....                      | 18        |
| File Selection.....                       | 18        |
| Detection Modes & Thresholds.....         | 18        |
| Timing & Envelope Analytics.....          | 18        |
| Analysis Filters.....                     | 19        |
| Export Formatting.....                    | 19        |
| Generator Shruti.....                     | 19        |

|                                             |    |
|---------------------------------------------|----|
| Introduction.....                           | 19 |
| Controls.....                               | 20 |
| Presets.....                                | 20 |
| Generator Tone.....                         | 20 |
| Introduction.....                           | 20 |
| Waveform & Routing.....                     | 20 |
| Frequency Control & Sweeps.....             | 21 |
| Volume.....                                 | 21 |
| Playback & Pulsate (Cable Tester Mode)..... | 21 |
| Generator Wind.....                         | 22 |
| introduction.....                           | 22 |
| Global & Routing Controls.....              | 22 |
| The Filter Cascade.....                     | 22 |
| Interface Interaction.....                  | 23 |
| GUI Knob Maker.....                         | 23 |
| Introduction.....                           | 23 |
| Interface Overview.....                     | 23 |
| Project Settings.....                       | 24 |
| Working with Layers.....                    | 24 |
| Layer Parameters.....                       | 24 |
| Appearance.....                             | 24 |
| Shape Properties.....                       | 25 |
| Base Size and Anchor.....                   | 25 |
| Repeater (Radial Array).....                | 25 |
| Outer & Inner Shadows.....                  | 25 |
| Blur.....                                   | 26 |
| Texture, Pattern, Grain.....                | 26 |
| Gradients.....                              | 26 |
| Animation (Start → End).....                | 26 |
| Previewing & Exporting.....                 | 27 |
| GUI Sheet Maker.....                        | 27 |
| Introduction.....                           | 27 |
| Interface.....                              | 27 |
| Source Images.....                          | 27 |
| Preview & Settings.....                     | 28 |
| Preview Mode.....                           | 28 |
| Sheet Settings.....                         | 28 |
| Playback & Export.....                      | 28 |
| GUI Sheet Slicer.....                       | 29 |
| Introduction.....                           | 29 |
| Loading a Sheet.....                        | 29 |

|                                          |    |
|------------------------------------------|----|
| Setting the Grid.....                    | 30 |
| The Divisibility Rule.....               | 30 |
| Previewing Before You Commit.....        | 30 |
| Auto-Trim.....                           | 31 |
| Slicing.....                             | 31 |
| Presets.....                             | 31 |
| Kontakt NCKP Info Pane.....              | 32 |
| Introduction.....                        | 32 |
| How to Use.....                          | 32 |
| Kontakt NCKP KSP Persistent.....         | 32 |
| Introduction.....                        | 32 |
| How to Use.....                          | 33 |
| Parsing Options.....                     | 33 |
| Kontakt NKS2 Mapper.....                 | 33 |
| Introduction.....                        | 33 |
| Groups & Visualizer.....                 | 34 |
| The Configuration Table.....             | 34 |
| Generated KSP & Presets.....             | 35 |
| Kontakt Performance View.....            | 35 |
| Introduction.....                        | 35 |
| Global Workflow & Top Controls.....      | 35 |
| The NCKP Tree (Left Pane).....           | 36 |
| Context Menu (Right-Click).....          | 36 |
| Keyboard Shortcuts.....                  | 36 |
| The Property Inspector (Right Pane)..... | 37 |
| The Preview Window (WYSIWYG Canvas)..... | 37 |
| Canvas Navigation Bar.....               | 37 |
| Edit Mode.....                           | 37 |
| Play Mode.....                           | 38 |
| MIDI 8 Controls.....                     | 38 |
| Introduction.....                        | 38 |
| Global Actions.....                      | 38 |
| The Mixer Channels.....                  | 38 |
| Utilities.....                           | 39 |
| MIDI Generator Nord Drum.....            | 39 |
| Introduction.....                        | 39 |
| Output Settings.....                     | 39 |
| Styles Selection.....                    | 40 |
| Generator Parameters.....                | 40 |
| Presets & Generation.....                | 41 |
| Under the Hood: Technical Details.....   | 41 |

|                                               |    |
|-----------------------------------------------|----|
| MIDI Keys.....                                | 41 |
| Introduction.....                             | 41 |
| Interface.....                                | 41 |
| Keyboard Settings.....                        | 41 |
| The Virtual Keyboard.....                     | 42 |
| Utility Controls.....                         | 42 |
| MIDI Monitor.....                             | 42 |
| Introduction.....                             | 42 |
| Options.....                                  | 42 |
| Event Filters.....                            | 42 |
| Utilities.....                                | 43 |
| MIDI XY Bowl.....                             | 43 |
| Introduction.....                             | 43 |
| The Bowl.....                                 | 44 |
| Mode.....                                     | 44 |
| Weight and Friction.....                      | 44 |
| Driving the Bowl.....                         | 44 |
| MIDI Lanes.....                               | 45 |
| Touch and Randomize.....                      | 45 |
| Presets.....                                  | 45 |
| MIDI to OSC Sender.....                       | 46 |
| Introduction.....                             | 46 |
| Interface & Controls.....                     | 46 |
| Target OSC Server.....                        | 46 |
| MIDI to OSC Routing Matrix.....               | 46 |
| Utility Controls.....                         | 47 |
| NKS Controller Mapper.....                    | 47 |
| Introduction.....                             | 47 |
| The User Interface.....                       | 47 |
| Writing a Controller Map (The Syntax).....    | 47 |
| The Basic Format.....                         | 47 |
| The Rules of the Mapper.....                  | 48 |
| Advanced Mapping: Pages and Blank Spaces..... | 48 |
| Creating Multiple Pages.....                  | 48 |
| Leaving Blank Spaces (Skipping Knobs).....    | 48 |
| NKS Create Preset.....                        | 48 |
| Introduction.....                             | 49 |
| Interface Overview.....                       | 49 |
| Patch Source Settings.....                    | 49 |
| Navigation Settings (Nav Mode).....           | 49 |
| Maschine Macro Settings (GUI Automation)..... | 49 |

|                                                      |    |
|------------------------------------------------------|----|
| Timing Settings (Crucial for Stability).....         | 50 |
| NKS Preset Tagging.....                              | 50 |
| Introduction.....                                    | 50 |
| The Interface.....                                   | 50 |
| How It Works: The Lexical Engine (Text Parsing)..... | 51 |
| Auto Tag Controls.....                               | 51 |
| Properties Tab (Metadata Writer).....                | 52 |
| Committing and Clearing Changes.....                 | 52 |
| NKS Sample Tagging.....                              | 52 |
| Introduction.....                                    | 52 |
| The Interface.....                                   | 53 |
| How It Works: Audio & Loop Parsing.....              | 53 |
| Auto Content Type.....                               | 53 |
| Properties Tab (Metadata Writer).....                | 54 |
| Committing and Clearing Changes.....                 | 54 |
| OSC 8 Controls User Manual.....                      | 54 |
| Introduction.....                                    | 54 |
| Interface & Controls.....                            | 55 |
| Target OSC Server.....                               | 55 |
| Global Actions.....                                  | 55 |
| OSC Controllers (Channel Strips).....                | 55 |
| Utility Controls.....                                | 56 |
| OSC Monitor.....                                     | 56 |
| Introduction.....                                    | 56 |
| A Note on OSC Networking.....                        | 56 |
| Interface & Controls.....                            | 56 |
| Network Connection.....                              | 56 |
| Filtering & Display Options.....                     | 57 |
| The Log Display.....                                 | 57 |
| Utility Controls.....                                | 57 |
| OSC XY Bowl.....                                     | 57 |
| Introduction.....                                    | 57 |
| The Value Model.....                                 | 58 |
| OSC Lanes.....                                       | 59 |
| Touch and Randomize.....                             | 59 |
| Presets.....                                         | 59 |
| OSC to MIDI Server.....                              | 60 |
| Introduction.....                                    | 60 |
| Quick Start Guide.....                               | 60 |
| Interface Breakdown.....                             | 60 |
| Server Settings.....                                 | 60 |

|                                               |    |
|-----------------------------------------------|----|
| The 8-Channel Mixer.....                      | 61 |
| Presets and Routing Recall.....               | 61 |
| Troubleshooting.....                          | 61 |
| Sampler Auto MIDI.....                        | 62 |
| Introduction.....                             | 62 |
| File Controls.....                            | 62 |
| MIDI Controls.....                            | 62 |
| Program Controls (Batch Sampling).....        | 62 |
| Audio & Process Controls.....                 | 63 |
| Recording Controls.....                       | 63 |
| Main Actions & Presets.....                   | 63 |
| Sampler Auto Threshold.....                   | 64 |
| Introduction.....                             | 64 |
| File Settings.....                            | 64 |
| Audio Settings (The Threshold Engine).....    | 64 |
| Process Settings.....                         | 64 |
| Recording Settings.....                       | 65 |
| Workflow Guide.....                           | 65 |
| Sampler Batch Mixer.....                      | 66 |
| Introduction.....                             | 66 |
| Interface.....                                | 66 |
| Source Controls (1-3).....                    | 66 |
| Processing Settings.....                      | 66 |
| Batch & Execution.....                        | 66 |
| Presets.....                                  | 67 |
| Sampler Batch Resonator.....                  | 67 |
| Introduction.....                             | 67 |
| The Core Concept.....                         | 67 |
| Source Configuration & Pre-Filtering.....     | 67 |
| Convolution Parameters.....                   | 68 |
| Processing & MIR Loop Extraction.....         | 68 |
| Batch Generation.....                         | 69 |
| Toolbox Music Handyman.....                   | 69 |
| Introduction.....                             | 69 |
| Interface.....                                | 69 |
| Music, Pitch, and Cycle Calculators.....      | 69 |
| Primary Outputs:.....                         | 70 |
| Advanced Dashboard:.....                      | 70 |
| Pitch, Tuning, and Harmonics.....             | 70 |
| Musical Scales, Chords, and Progressions..... | 71 |
| Toolbox RIFF Reader.....                      | 71 |

|                                            |    |
|--------------------------------------------|----|
| Introduction.....                          | 71 |
| File Navigation.....                       | 71 |
| Display Modes.....                         | 72 |
| The Display Tree.....                      | 72 |
| Presets.....                               | 72 |
| Toolbox Sampleset Check.....               | 73 |
| Introduction.....                          | 73 |
| How to Use.....                            | 73 |
| Smart Scanning: Fast vs. Deep Scan.....    | 73 |
| The Checks & Rulesets.....                 | 73 |
| Deviance Checks (The Two-Pass Method)..... | 73 |
| Structural Integrity.....                  | 74 |
| Logic Checks.....                          | 74 |
| Toolbox Smart Renamer.....                 | 74 |
| Introduction.....                          | 74 |
| Interface & Controls.....                  | 74 |
| Targeting & Filtering.....                 | 74 |
| Operation Logic.....                       | 75 |
| Positional Conditions.....                 | 75 |
| Casing Targets.....                        | 76 |
| The Preview Engine.....                    | 76 |
| Color Coding & Collision Detection.....    | 76 |
| Execution & Output.....                    | 76 |
| Presets.....                               | 77 |
| Wavetable Converter.....                   | 77 |
| Introduction.....                          | 77 |
| Process.....                               | 77 |
| Controls.....                              | 78 |
| Wavetable Creator.....                     | 78 |
| Introduction.....                          | 78 |
| Waves.....                                 | 79 |
| Wave Shapes.....                           | 79 |
| Sine.....                                  | 80 |
| Tri.....                                   | 80 |
| Saw.....                                   | 80 |
| Square.....                                | 81 |
| Noise.....                                 | 81 |
| Harmonics.....                             | 82 |
| Randoms.....                               | 82 |
| Special.....                               | 83 |
| Wave Shapers.....                          | 84 |

|                                 |     |
|---------------------------------|-----|
| Off.....                        | 84  |
| Filters.....                    | 84  |
| Shift.....                      | 84  |
| Shapers.....                    | 84  |
| Decimators.....                 | 85  |
| Special.....                    | 85  |
| Waves Interaction.....          | 86  |
| Interaction Types.....          | 86  |
| Off.....                        | 86  |
| Crossfades.....                 | 86  |
| Mixing.....                     | 87  |
| Modulations.....                | 87  |
| Special.....                    | 87  |
| Random.....                     | 88  |
| Wavetable Creation.....         | 89  |
| Export Settings.....            | 89  |
| Creation.....                   | 89  |
| Player.....                     | 90  |
| Visualizer.....                 | 90  |
| Presets.....                    | 91  |
| Compatibility.....              | 91  |
| Wavetable Bi-Cycle Morpher..... | 94  |
| Introduction.....               | 94  |
| Anchors.....                    | 95  |
| Shapers.....                    | 95  |
| Interaction.....                | 95  |
| Random.....                     | 96  |
| Wavetable Creation.....         | 96  |
| Export Settings.....            | 96  |
| Creation.....                   | 97  |
| Player.....                     | 97  |
| Visualizer.....                 | 97  |
| Presets.....                    | 98  |
| Wavetable Cycle Creator.....    | 98  |
| Introduction.....               | 98  |
| Source.....                     | 99  |
| Wave Shapes.....                | 99  |
| Wave Shapers.....               | 99  |
| Random.....                     | 99  |
| Single Cycle Creation.....      | 100 |
| Export Settings.....            | 100 |

|                                   |            |
|-----------------------------------|------------|
| Creation.....                     | 100        |
| Visualizer.....                   | 101        |
| Presets.....                      | 101        |
| Wavetable Extractor.....          | 101        |
| Introduction.....                 | 101        |
| Source Audio.....                 | 101        |
| Extraction Pipeline Settings..... | 102        |
| Core Analysis.....                | 102        |
| Spectral Shaping.....             | 102        |
| Detail Sliders (0 - 100%).....    | 103        |
| Presets.....                      | 103        |
| Actions.....                      | 103        |
| Wavetable Preview.....            | 103        |
| Wavetable Viewer.....             | 104        |
| Introduction.....                 | 104        |
| File Navigation & Parsing.....    | 104        |
| Display Modes.....                | 104        |
| Presets.....                      | 105        |
| Zeta Ocean Quiz.....              | 105        |
| Introduction.....                 | 105        |
| How to Play.....                  | 105        |
| Scoring System.....               | 105        |
| Zeta Xaxixot.....                 | 106        |
| Introduction.....                 | 106        |
| Controls.....                     | 106        |
| The Grid & Hazards.....           | 106        |
| Gameplay Mechanics.....           | 106        |
| Pickups.....                      | 106        |
| Interface & Settings.....         | 107        |
| Zeta Xong.....                    | 107        |
| Introduction.....                 | 107        |
| Controls.....                     | 107        |
| Interface & Settings.....         | 107        |
| <b>Change Log.....</b>            | <b>107</b> |
| Version 1.10.x.....               | 107        |
| Version 1.2.0.....                | 108        |
| Version 1.0.0.....                | 108        |
| Version 0.13.21.....              | 108        |
| Attribution.....                  | 108        |

# Introduction

Welcome to the Sound Creator Suite. At Ocean Swift Synthesis, our goal is to create innovative sounds and instruments that elevate the music production and sound design experience. This suite provides a robust, centralized host environment for running our specialized tools and utilities.

Process audio in bulk, align loops, generate MIDI, build and convert wavetables, tag NKS presets and samples, run precision tone tests, and access instant music-calculation tools from one powerful suite.

## Navigation and Tool Selection

The top navigation bar is your primary control center:

- **Tool Selector:** Use the dropdown menu to instantly switch between active Ocean Swift tools (e.g., the default "**Alpha Station**" tool).
- **Last:** Click the **Last** button to instantly jump back to the previously loaded tool. This allows you to rapidly toggle back and forth between your two most recently used tools, and will even remember your last active tool from a previous session when you launch the application.
- **Settings:** Click the **Settings** button to open the configuration overlay. The application will remember your active tool seamlessly in the background while you adjust your preferences.

## Settings and Configuration

Proper audio routing is essential for accurate sound generation. Click the **Settings** button to access these parameters:

### Audio and MIDI Settings

- **Audio Output Device:** Select the hardware output device for sound generation. The suite automatically queries your system to find valid active outputs.
- **Audio Input Device:** Select the hardware input device used for sampling and recording audio.
- **Audio Buffer Size:** Set the buffer size for audio processing. Lower values reduce latency, giving you a tighter response, but they may cause audio crackling on slower systems.
- **MIDI Input Port:** Select the port for routing an external MIDI controller into the suite to trigger sounds.
- **MIDI Output Port:** Select the port for sending MIDI data out from the application to external hardware or software.

# Kontakt Developer Integration

This section provides powerful workflow automation specifically designed for Native Instruments Kontakt instrument developers.

- **Kontakt Executable Path:** Specify the full system path to the standalone Native Instruments Kontakt executable (e.g., **Kontakt 8.app** on macOS or **Kontakt 8.exe** on Windows). Use the **Browse...** button to easily locate it on your system.
- **Launch / Stop Kontakt:** Click to launch Kontakt as an isolated child process. When running, the suite will capture Kontakt's internal command-line output and display warnings, errors, and info directly in the Log Viewer. The status indicator will display a green **Connected** or red **Disconnected** state.
- **Create NKS Instrument Structure:** Automates the boilerplate setup of a Native Kontrol Standard (NKS) library. When clicked, it prompts you for a destination folder and a Product Name. The suite will instantly generate the strict NKS folder hierarchy (including **unencoded/Artwork, Samples, Snapshots**, etc.) and silently execute a background Lua script through Kontakt to generate and save your base **.nki** instrument file and link it to an **.nkr** container.
  - **Keep Alive:** When checked, keeps the Kontakt instance open and connected to the application logger when creating an NKS instrument structure. Otherwise, the Kontakt instance is closed when the creation process is complete.

## Application Preferences

- **Online Manual:** Instantly opens this official Sound Creator Suite user manual in your default web browser.
- **Enable Tooltips:** Toggle the display of helper tooltips that appear when hovering over buttons and controls. Disable this once you are familiar with the interface to reduce visual clutter.
- **Check on Launch:** If checked, the suite will automatically ping the Ocean Swift servers when the application starts to see if a newer version is available.
- **Check for Updates:** Manually checks if your currently running version is up to date. If a new version is found, a download link and notification will be printed in the Log Viewer.

## The Log Viewer

The bottom panel of the suite is a dedicated, thread-safe Log Viewer capable of maintaining a 10,000-line history of application status, background operations, and system errors.

### Log Controls

- **Clear:** Instantly wipes the current log history from both the UI and the underlying **.log** file.
- **Copy:** Copies the raw text of the current log history directly to your system clipboard.
- **Save Log...:** Exports the log history to a **.txt** file for external review or bug tracking.

## Log Filtering

When hunting for specific information during tool operation, use the filtering checkboxes located on the right side of the log panel:

- **Info / Warn / Error:** Check or uncheck these boxes to show or hide specific severity levels on the fly. This does not delete the hidden logs; it simply removes them from the current view.
- **Color:** Check this box to enable severity-based syntax highlighting. **INFO** messages will display in Ocean Swift green, **WARN** messages in yellow, and **ERROR** messages in red.

*Note: All of your settings, including your log filter preferences and color toggle, are automatically saved and will be restored the next time you launch the Sound Creator Suite.*

## Safe Shutdown Policy

If you attempt to close the Sound Creator Suite while a tool is actively running a process, the application will intercept the closure to protect your data. You will be prompted with a warning dialog offering two choices:

- **Cancel operation & Quit:** Safely requests the active tool to halt its process before shutting down.
- **Don't quit:** Returns you to the application.

If a cancellation request hangs for more than 6 seconds, a secondary failsafe will offer a force quit option to prevent the application from freezing.

## The Tools

### Overview

- **Alpha Station** - The entry point to the Sound Creator Suite.
- **Audio Batch Process** - An offline DSP processor for applying signal chains to thousands of audio files simultaneously.
- **Audio Loop Aligner** - Leverages MIR algorithms to analyze and perfectly remove silence at the start of audio files.
- **Generator Shruti** - A small digital adaptation of a traditional Indian Shruti box.
- **Generator Tone** - A precision utility for generating pure, high-fidelity test tones, sweeps, and noise profiles.
- **Generator Wind** - A generative sound design instrument built specifically to synthesize dynamic noise sweeps.
- **GUI Knob Maker** - Designed to generate UI components for audio plugins, and Kontakt instruments.

- **GUI Sheet Maker** - Take sequences of individual image frames and seamlessly stitch them together into a single sprite sheet.
- **GUI Sheet Slicer** - Cuts an existing sprite sheet back into its numbered individual frames, the exact inverse of the Sheet Maker.
- **Kontakt NCKP Info Pane** - Parses Native Instruments .nckp files to extract and compile Info Pane documentation into a readable list.
- **Kontakt NCKP KSP Persistent** - designed to automate translating and writing NCKP to KSP persistence declarations.
- **Kontakt NKS2 Mapper** - A visual authoring environment designed to create KSP script sections for NKS2 controller mapping.
- **Kontakt Performance View** - An advanced take on Kontakt performance view editing.
- **MIDI 8 Controls** - Provides a clean, 8 channel mixer interface for sending MIDI CC messages.
- **MIDI Generator Nord Drum** - An algorithmic sequencer built to generate MIDI patterns tailored specifically to the Nord Drum hardware.
- **MIDI Keys** - A responsive, built-in virtual keyboard designed for rapid sound design auditioning.
- **MIDI Monitor** - Inspect and analyze the MIDI data in real-time.
- **MIDI XY Bowl** - A physics-driven XY pad where a ball rolling in a bowl sends MIDI CC across six lanes at once.
- **MIDI to OSC Sender** - Dedicated routing matrix that bridges your physical MIDI controllers with OSC environments.
- **NKS Create Preset** - An automation utility designed to batch export presets from software synthesizers hosted in Maschine as NKS presets.
- **NKS Controller Mapper** - Defines a single, perfect hardware control layout for a synthesizer by a very simple declarative script.
- **NKS Preset Tagging** - A batch metadata injection tool designed to prepare software synthesizer presets for the Native Instruments hardware ecosystem. Easily categorize your patches for seamless, structured browsing in Maschine and Complete Kontrol.
- **NKS Sample Tagging** - A batch tagger that safely embeds proprietary Native Instruments NKS properties directly into [.wav](#) files.
- **OSC 8 Controls** - A customizable, 8-channel virtual mixer that sends Open Sound Control (OSC) data across your network.
- **OSC Monitor** - Diagnostic tool for viewing incoming Open Sound Control (OSC) messages over your network.
- **OSC XY Bowl** - The OSC counterpart to the MIDI XY Bowl, streaming a rolling ball's motion as Open Sound Control across six lanes.
- **OSC to MIDI Server** - A lightweight background server that translates Open Sound Control (OSC) network traffic into routable MIDI messages.
- **Sampler Auto MIDI** - An automated MIDI engine designed to streamline hardware sampling.
- **Sampler Auto Threshold** - An automated listening and capturing engine designed to streamline hardware sampling.

- **Sampler Batch Mixer** - The Sampler Batch Mixer combines up to three audio sources with independent highpass and lowpass filtering.
- **Sampler Batch Resonator** - Mathematically imprint the acoustic and tonal characteristics of one audio folder onto another.
- **Toolbox Music Handyman** - A rapid-access collection of sound and music related calculators. Instantly calculate delay times, tempo-to-millisecond conversions, and note-to-frequency math.
- **Toolbox RIFF Reader** - A diagnostic inspector for .wav file headers.
- **Toolbox Sampleset Check** - Recursively scans directories of .wav files and flags any assets that deviate from the expected standard.
- **Toolbox Smart Renamer** - batch file renaming utility designed to handle massive libraries of samples, presets, and project folders.
- **Wavetable BiCycle Morpher** - Loads two single-cycle waveforms and morphs between them across a full wavetable using the complete interaction palette.
- **Wavetable Converter** - Seamlessly translates existing wavetable audio files between different software synthesizer formats.
- **Wavetable Creator** - A powerful generation environment for building custom wavetables from scratch.
- **Wavetable Cycle Creator** - Builds single-cycle waveforms from a palette of shapes and math processors, chained through multiple shaper stages.
- **Wavetable Extractor** - Uses spectral information from a loaded audio file to construct a wavetable.
- **Wavetable Viewer** - Diagnostic tool for inspecting and visualizing .wav based wavetables.
- **Zeta Ocean Quiz** - A lightweight, built-in trivia mini-game designed specifically for audio engineers, producers, and synthesizer enthusiasts.
- **Zeta Xaxixot** - A high-stakes, territory-capture arcade game.
- **Zeta Xong** - Vertical homage to the classic arcade game of table tennis.

## Audio Batch Process

### Introduction

The Batch Process Tool is a mastering-grade, offline DSP processor designed specifically for sound designers, sample pack creators, and virtual instrument developers. It allows you to recursively scan a folder of audio files (.wav, .aif, .aiff) and apply a rigorous, mathematically precise chain of processing blocks.

Whether you are preparing thousands of Wavestate multi-samples, formatting drum hits, or mangling raw oscillators into massive cinematic textures, this tool processes everything

perfectly and outputs to a clean **Processed** subfolder—leaving your original files completely untouched.

## Global Workflow & Presets

- **Select Folder:** Choose the root directory containing your source audio. The tool will recursively find all valid audio files within this folder and its subfolders.
- **Execution:** Clicking **Execute** runs the process on a safe background thread. Progress is displayed on the bottom bar, and the output is replicated in the main application log.
- **Output:** Processed files are saved to a new folder named **Processed** inside your selected root directory. The original folder hierarchy is perfectly preserved.
- **Presets:**
  - \* **Save Preset:** Saves your exact current UI/DSP configuration to a **.json** file. *(Note: Presets smartly ignore your current folder path so they can be loaded universally on any new project).*
  - **Load Preset:** Instantly recalls your custom DSP chains (e.g., "Lo-Fi Drum Crunch" or "Sci-Fi Pad Maker").

## Spectral & Pitch Processing

*This section alters the fundamental pitch, frequency, and stereo width of the source audio. It is the first stage in the DSP signal flow.*

- **Tape Pitch Shift:** A high-quality polyphase resampler. Shifts the pitch up or down (-60 to +60 semitones) by altering the speed of the audio, identical to classic tape machine varispeed.
- **Freq Shift (Sci-Fi):** Uses a mathematical Hilbert Transform to shift all frequencies by a fixed Hz amount (-5000 to +5000 Hz). This destroys the natural harmonic integer ratios, resulting in metallic, robotic, or bell-like textures.
- **Ring Mod:** Multiplies the audio by a pure sine wave carrier (0.1Hz to 20kHz). This creates upper and lower sidebands while erasing the fundamental pitch. Perfect for aggressive, "Dalek-style" cinematic impacts.
- **Unison Doubler:** Instantly transforms thin mono oscillators into massive "Supersaw" patches.
  - *How it works:* It forces the audio to a stable mono anchor, generates two identical voices pitched slightly up and down, and pans them wide.
  - *Safety:* Includes built-in equal-power (-3dB) gain compensation to guarantee the stereo mix will never internally clip.

---

## Phase & Utility

*Essential clean-up tools to prepare the waveform for dynamic processing.*

- **Reverse Audio:** Flips the audio array backward. (Applied *after* pitch/spectral shifting).
- **Invert Phase:** Flips the polarity of the waveform (multiplies by -1.0).
- **Kill DC:** Removes DC offset by subtracting the exact mean value of the array per-channel. Essential for analog hardware recordings.
- **Kill Hiss (18kHz):** Applies a transparent, zero-phase 4th-order Butterworth low-pass filter at 18kHz. Gently removes digital aliasing or analog air without smearing transient punch.

## Time & Amplitude Envelopes

*Precision timing operations to guarantee click-free playback in samplers like Kontakt.*

- **Trim Start Silence:** Automatically crops digital silence and noise floors from the start of the file. Selectable thresholds from **-80dB** to **-20dB**.
- **Trim End Silence:** Automatically crops digital silence and noise floors from the tail of the file. Selectable thresholds from **-80dB** to **-20dB**.
- **Fade In & Fade Out:** Applies volume fades to the start and end of the file.
  - *Duration:* Configurable from 1ms up to 10 seconds.
  - *Curve Shapes:* Choose between **Linear**, **Exponential** (great for natural cymbal/reverb decays), **Logarithmic** (great for quick pad swells), or **S-Curve** (smoothstep).

## Saturation & Analog Character

*Applied after fades, ensuring that the volume shape is uniformly saturated and flattened.*

- **Tube Exciter (Even Harmonics):** Mathematically squares the waveform to generate warm, thick, 2nd-order harmonics (sub-bass thickening), blending it back into the original signal.
- **Soft Clip / Overdrive (Odd Harmonics):** Applies a smooth, analog-style hyperbolic tangent (**tanh**) curve. Shaves off harsh digital transients and replaces them with punchy, tape-like overdrive. Drive ranges from 1.0x to 10.0x.
- **Symmetrize Phase (Maximize Headroom):** A mastering engineer's secret weapon. Uses a cascade of all-pass filters to rotate the phase of asymmetrical waveforms (like raw analog PWM synths, brass, or vocals) until they are visually symmetrical. This allows the file to be normalized 2 to 3 dB louder.
  -  **WARNING:** Do not use this on percussive material or drum hits, as phase rotation will smear the transient click.

## Leveling & Output Formatting

*The final stage of the engine, ensuring your files are perfectly optimized for delivery.*

- **Peak Normalize:** Scales the audio to hit an exact dBFS target (e.g., **-6.0 dB**).

- *Per Channel*: If checked, Left and Right are normalized independently (fixes lopsided stereo files). If unchecked, channels are linked to preserve the original stereo balance and panning.
- **Sample Rate Convert**: Select a new target sample rate (e.g., 44.1kHz, 48kHz, 96kHz). Uses high-fidelity polyphase resampling.
- **Bit Depth Convert**: Target 16-bit or 24-bit PCM output.
- **Dither**: Applies high-quality TPDF (Triangular Probability Density Function) dither. *Highly recommended* if you are down-sampling to 16-bit or if you have engaged any of the DSP stages (Pitch, SR conversion, EQ, Fades).

## Loop Detection

*Automatically detect and inject seamless loop points using advanced Music Information Retrieval.*

- **Find Loop (WAV only)**: Analyzes the audio to find the optimal seamless loop points and automatically embeds them as SMPL chunk data directly into the WAV file header.
  - **Ignore Attack**: Skips a defined percentage of the initial audio (e.g., 20%) to prevent the loop from capturing transient strikes or non-sustaining start characteristics.
  - **Ignore Tail**: Defines the cut-off percentage near the end of the file (e.g., 80%) to avoid looping the natural decay, reverb tails, or digital silence.
  - **Min Loop Time**: Enforces a minimum duration for the extracted loop in milliseconds (default 500ms) to ensure rhythmic or tonal stability.
  - **Analysis Quality**: Determines the resolution (hop size) of the MIR feature extraction. *High* offers the finest precision for finding the perfect phase match, while *Fast* significantly reduces processing time for larger batches.
  - **Crossfade**: Applies a 10ms crossfade between the loop end and start points.
  - **Crop Loop**: Deletes all audio outside of the detected loop, exporting only the looping section.

## Internal Signal Flow Order

Understanding the internal order of operations helps you predict the result of your batches. The DSP chain processes in this exact order:

1. **Read Array** (Float32 to prevent internal clipping)
2. **Pitch / Freq / Ring Mod / Unison** (Spectral generation)
3. **Reverse / Phase Invert / DC Kill / Hiss Kill** (Utility & cleanup)
4. **Trim Silence** (Length reduction)
5. **Fade In / Fade Out** (Volume shaping)
6. **Tube Exciter / Soft Clipper** (Saturation)

7. **Phase Symmetrizer** (Headroom prep)
8. **Peak Normalization** (Final leveling)
9. **Sample Rate / Bit Depth Conversion + Dither**
10. **Find Loop**
11. **Write to Disk**

## Audio Loop Aligner

### Introduction

The MIR (Music Information Retrieval) Loop Aligner is a precision batch-processing utility designed to solve one of the most common headaches in sample pack creation and music production: micro-silences at the beginning of audio files.

When bouncing stems from a DAW or recording hardware synthesizers, there is almost always a tiny gap of silence or background noise before the actual musical transient hits. If left uncorrected, these loops will drift off-grid when placed in a sequencer. The MIR Loop Aligner mathematically analyzes your audio, detects the exact sample of the first true musical hit, and surgically trims the file so it lands perfectly on the grid—all in bulk, and without altering the sonic character of your original audio.

### Universal Format Engine

The Aligner features a lossless, universal I/O engine. Whether your loops are standard 16-bit Mono, 24-bit Stereo, or even 32-bit float Surround sound at 96kHz, the tool will natively read, process, and export the file in its **exact original format and bit-depth**.

*Note: For multi-channel files (like stereo), the engine temporarily sums the audio to mono strictly in the background to calculate the perfect cut point, and then applies that exact cut across all pristine original channels simultaneously.*

### Interface Guide

#### File Selection

- **Base Folder:** Select the parent directory containing your audio. The tool will recursively scan this folder (and all subfolders) for `.wav` files.
- **Output:** Processed files are automatically saved into a new `_trimmed` folder created next to your original files, preserving your original folder hierarchy. You will never overwrite your source audio.

#### Detection Modes & Thresholds

These settings determine how the algorithm "listens" to your audio to find the first hit.

- **Analysis Mode: \* HPF+LPF Envelope (Recommended):** Uses a smoothed volume envelope and frequency filters to ignore low-end rumble and high-end hiss, focusing strictly on the musical transient.
  - **HPF+LPF Peak:** Looks for raw, instantaneous volume spikes within the filtered audio.
  - **Raw Envelope / Raw Peak:** Disables the filters and analyzes the raw audio signal.
- **Silence Threshold:** The volume level (in dBFS) that the audio must cross to be considered a "hit." If the tool is trimming too early because of background noise, raise this value (e.g., from -50.0 to -40.0).

### Timing & Envelope Analytics

These parameters fine-tune the algorithm to prevent false positives (like a breath or a quiet click before a drum hit).

- **Envelope Smoothing (ms):** Determines how smooth the analysis envelope is. Higher values prevent the detector from reacting to twitchy, microscopic clicks.
- **Hold Time (ms):** The signal must stay above the Silence Threshold for this duration to trigger a detection. This prevents ultra-short electrical pops from being identified as a kick drum.
- **Backtrack (ms):** *This is the magic of the algorithm.* Once the tool is confident a real hit has occurred (by satisfying the Hold Time), it "backtracks" into the immediate past to find the exact microscopic sample where the transient began rising from silence.
- **Coarse Window (Peak modes only):** The chunk size used when scanning for instant peak crossings.

### Analysis Filters

These filters isolate the core frequencies of a transient (like the punch of a snare) so the detector isn't fooled by DC offset, sub-rumble, or high-frequency hiss.

- **HPF Cutoff:** Ignores frequencies below this value during analysis.
- **LPF Cutoff:** Ignores frequencies above this value during analysis.
- **Crucial Note:** These filters are **Analysis-Only**. They are used by the detector's "ears" to find the cut point, but they are absolutely never applied to your final exported audio.

### Export Formatting

- **Pre-roll before hit (ms):** If you don't want the audio chopped exactly at the zero-crossing of the transient, you can leave a few milliseconds of "breathing room" before the hit.
- **Fade-in (optional):** Applies a microscopic, mathematically perfect linear volume fade to the very beginning of the exported file. Set this to 1.0 or 2.0 ms to guarantee zero speaker clicks when the loop loops.

- **Preserve original length (pad end):** If checked, the tool calculates exactly how many samples it cut from the front of the audio, and adds that exact amount of digital silence to the very end of the file. This ensures your 2-bar loop remains exactly 2 bars long for hardware samplers.

## Generator Shruti

### Introduction

Generator Shruti is a digital adaptation of a traditional Indian Shruti box, designed for creating continuous drone textures. It utilizes pre-calculated algorithmic resampling to ensure artifact-free audio regardless of voice density.

### Controls

- **Voice Keys (1-13):** 13 chromatic toggle switches spanning one full octave. Activate multiple keys simultaneously to build custom drone chords and clusters.
- **Sample Select:** Selects between the available source samples.
- **Gender Switch:** Shifts the global root pitch of the instrument.
  - **Male (Off):** Lower octave register.
  - **Female (On):** Higher octave register.
- **Play / Stop:** Engages or mutes the master audio engine.
- **Volume:** Master output level.

### Presets

- **Save / Load:** Stores the state of the selected sample, 13 keys, the Gender switch, and the Volume level to a file.

## Generator Tone

### Introduction

The Tone Generator is a professional-grade, precision audio oscillator built directly into the Sound Creator Suite. Designed for both DSP developers and studio engineers, it provides mathematically pure waveforms, acoustic room-testing tools, and diagnostic cable-testing features.

Every function in the Tone Generator is heavily protected by anti-pop smoothing filters and phase-continuous algorithms, ensuring that rapid parameter changes will never damage your hearing or your studio monitors.

### Waveform & Routing

- **Waveform Selector:** Choose the fundamental shape of your test signal.
  - **Sine:** A pure fundamental frequency with no harmonics. Perfect for testing resonances and identifying specific acoustic reflections.
  - **Saw:** Contains all integer harmonics. Excellent for testing filter slopes and subtractive synthesis chains.
  - **Square:** Contains only odd harmonics. Great for testing distortion characteristics and pulse-width logic.
  - **White Noise:** Equal energy per frequency *bin*. Useful for raw digital system testing.
  - **Pink Noise:** Equal energy per *octave* (processed via a 3-pole IIR filter). This is the industry standard for calibrating studio monitors and testing human-perceived frequency response.
- **Routing:** Dynamically route the audio to your interface's **Left**, **Right**, or **Center** (Both) channels. This is crucial for isolating dead cables, troubleshooting stereo width issues, or feeding a specific hardware chain. *(Note: If you are using a mono-only output device, the tool will automatically sum to mono regardless of this setting).*

## Frequency Control & Sweeps

The Tone Generator features a highly responsive, click-free frequency engine that allows you to sweep across the human hearing range (20 Hz to 20,000 Hz) in real-time.

- **Precise Input (SpinBox):** Type in an exact frequency down to a single decimal point.
- **Logarithmic Slider:** A finely-tuned logarithmic slider that mimics the feel of a premium hardware oscillator, preventing low frequencies from being cramped at the bottom of the dial.
- **Quick-Jump Buttons:**
  - **100:** Instantly jump to 100 Hz to check sub-bass and woofer response.
  - **1k:** The universal 1 kHz standard reference tone.
  - **10k:** Instantly jump to 10 kHz to check tweeter response and high-end clarity.
- **Automated Sweep:** Click **Sweep** to trigger an automated, mathematically perfect logarithmic glide from 20 Hz to 20 kHz.
  - Use the **Sweep Time Dropdown** to select the duration (1s, 5s, 10s, or 30s).
  - *Pro Tip:* Use a fast 1-second sweep for a quick system check, or a 30-second sweep to walk around your studio room and listen for standing waves or resonant rattling in physical hardware.

## Volume

- **Output Level:** Controls the final output amplitude of the tool.
- **Safety feature:** To prevent accidental digital clipping and protect your hardware, the absolute maximum output of the slider is internally scaled to a safe threshold (~-2.5 dBFS).

## Playback & Pulsate (Cable Tester Mode)

- **Play / Stop:** Initiates or stops the audio stream. Every play/stop action is cushioned by a 50-millisecond fade-in and fade-out envelope to prevent DC-offset popping.
- **Pulsate (ON/OFF):** Engages the "Cable Tester" mode. When active, the audio signal pulses on and off repeatedly. This gives you a clear reference signal that doesn't cause ear fatigue while you trace cables behind a desk or test microphone inputs.
- **Pulse Rate:** Choose the speed of the pulsation:
  - **100 ms:** Fast, machine-gun flutter.
  - **250 ms:** Standard fast pulse.
  - **500 ms:** Standard slow pulse (120 BPM).
  - **1000 ms:** 1 second ON, 1 second OFF.
  - **2000 ms:** 2 seconds ON, 2 seconds OFF (Ideal for testing reverb and delay tails in a room).

## Generator Wind

### introduction

Generator Wind is a generative sound design instrument built specifically to synthesize dynamic noise sweeps. It utilizes a continuous noise source fed through a cascade of three modulatable bandpass filters, each equipped with dedicated saturation and modulation routing.

### Global & Routing Controls

The top section governs the core engine, audio routing, and master output.

- **Play / Stop:** Engages or stops the audio engine. Playback includes a smooth 50ms amplitude fade to prevent speaker pops.
- **Input:** Selects the raw noise source fed into the first filter.
  - **Pink Noise:** Features equal energy per octave, resulting in a darker, more natural, and rumbling texture. Highly recommended for realistic wind and deep atmospheres.
  - **White Noise:** Features equal energy per frequency, resulting in a bright, hissy spectrum. Ideal for sharp, whistling gusts or synthetic noise sweeps.
- **Out Select:** Determines the final tap point of the audio chain. You can choose to listen to the output directly after Filter 1, after Filter 2, or at the very end of the chain after Filter 3.
- **Aux Rate:** Sets the frequency (0.001 Hz to 10.0 Hz) of the global Auxiliary Sine LFO. This single LFO can be routed to modulate all three filters simultaneously for synchronized, breathing movement.
- **Master Gain:** The final output volume of the module. *Important: To protect your hearing and equipment from unexpected volume spikes, the Master Gain parameter is explicitly excluded from the Preset save and recall system.*

## The Filter Cascade

The core of the sound design happens in the three Bandpass filter stages. Audio flows in series: Filter 1 feeds into Filter 2, which feeds into Filter 3. Each stage contains the following controls:

- **Pre-Gain:** Adjusts the input volume (-24 dB to +24 dB) before the audio hits the filter. Pushing the gain high allows you to drive the signal harder into the subsequent saturation stage.
- **Cutoff:** The base center frequency of the Bandpass filter (20 Hz to 20,000 Hz).
- **Res (Resonance):** Narrows the bandwidth of the filter (0% to 100%). Higher values emphasize the cutoff frequency, transforming broad noise into pitched, whistling, or howling tones.
- **LFO Mod:** Sets how much the filter's dedicated, independent LFO modulates the cutoff frequency. At 100%, the cutoff sweeps across a massive +/- 4 octave range.
- **Aux Mod:** Sets how much the global Auxiliary LFO modulates the cutoff frequency. You can combine this with the LFO Mod to create highly complex, unpredictable rhythmic movements.
- **Drive:** Applies a smooth, analog-style saturation (tanh soft-clipping) to the output of the filter. Higher values add harmonic grit and compress the dynamic range of high-resonance peaks.
- **LFO Rate:** Sets the speed (0.001 Hz to 10.0 Hz) of the dedicated Sine LFO for this specific filter stage.

## Interface Interaction

Every parameter in the filter cascade features a vertical slider for performative, tactile adjustments, paired with a high-precision digital readout box directly above it.

- **To make broad sweeps:** Click and drag the vertical slider.
- **To input exact values:** Click inside the text box, type your desired number (e.g., "440" for Cutoff, or "50" for Resonance), and press Enter. The tool will automatically format your input, clamp it to safe boundaries, and update the slider position instantly.
- **Save / Load Preset:** Save the entire state of the synthesizer (Input, Routing, Aux Rate, and all 3 Filter stages) to a file. Recalling a preset allows you to instantly jump between entirely different acoustic environments. (*Note: Master Gain is never recalled*).

## GUI Knob Maker

### Introduction

This tool is designed to generate studio-grade UI components for audio plugins and Kontakt instruments. Whether you need photorealistic brushed-metal knobs, glowing LED rings,

procedural film grain, or mechanical toggle switches, the GUI Knob Maker allows you to design and export complex animations as perfectly stitched sprite sheets in seconds.

## Interface Overview

The interface is divided into three primary workspaces:

- **Left Pane (Project & Layers):** Manage your master project settings, save/load project files, and control your layer stack.
- **Middle Pane (Layer Parameters):** The engine room. This is where you mathematically design, style, and animate the currently selected layer.
- **Right Pane (Preview & Export):** An interactive 60fps canvas where you can scrub through your animation frames and export the final sprite sheets.

## Project Settings

At the top left, you define the global boundaries of your component:

- **Width / Height:** The exact pixel dimensions of a *single frame* of your UI element.
- **Frames:** The total number of animation frames. (e.g., 31, 64, or 128 frames for smooth rotary knobs).
- **Oversample:** The internal rendering multiplier (1x, 2x, 4x, 8x). Use **4x or 8x** before exporting to achieve perfectly smooth, anti-aliased edges. *Note: High oversampling takes slightly longer to export but yields commercial-grade results.*
- **Global Light:** Sets a universal lighting angle (in degrees). Layers synced to this light will automatically adjust their gradients and drop shadows to match, making it incredibly easy to change the light source for your entire synth UI with one dial.

## Working with Layers

The GUI Knob Maker uses a Photoshop-style layer stack. The layer at the **top** of the list renders in the background, while the layer at the **bottom** of the list renders in the foreground.

- **[+] / [-]:** Add or delete layers.
- **Duplicate:** Instantly copy a complex layer to tweak it.
- **Up / Down:** Move layers backward or forward in the visual stack.
- **Context Menu (Right-Click):** Right-click any layer to **Save Layer Preset** or **Load Layer Preset**. This allows you to build up a library of reusable "Ocean Swift" materials (like your favorite knurled metal base) and drop them into future projects.

## Layer Parameters

When you select a layer, the middle pane populates with its exact mathematical parameters, divided into collapsible sections.

## Appearance

- **Shape:** Choose from mathematical primitives (Circle, Rectangle, Polygon, Value Arcs, Lines) or a custom Image. The **Metal Circle** shape auto-generates a complex, anisotropic conical gradient perfect for hardware knobs.
  - **Assets:** When using an Image or Texture, click this dropdown to browse a recursively organized library of high-quality, pre-installed factory assets (like Woods and Metals). Factory assets use portable tokens, meaning your preset files will flawlessly load on any other computer or OS!
- **Fill & Stroke:** Independently enable the solid interior (Fill) and the outline (Stroke).
  - *Pro Tip (Dynamic Inheritance):* If you turn **OFF** the Fill and turn **ON** the Stroke, your Gradients, Patterns, and Film Grain will automatically apply to the stroke line instead!
- **Line Style:** Change a solid stroke to Dashed, Dotted, or Dash-Dot.
- **Light Sync:** Check this to lock the layer's shadows and gradients to the Project's "Global Light" setting.
- **Clip Mask (Clip to Layer Below):** Check this to constrain the current layer to the exact physical boundaries of the layer beneath it. Perfect for keeping chaotic noise or wild gradients perfectly inside a base knob cap.
- **Blend Mode:** Standard photographic blending (Multiply, Screen, Overlay, Add, Darken).

## Shape Properties

Depending on your selected shape, specific mathematical controls will dynamically appear:

- **Sides / Count:** Defines the number of sides for a polygon, or the exact number of lines rendered in a **Lines Array**.
- **Arc Start / Rotation:** Sets the physical start angle of an Arc, or acts as the master orientation director for a **Lines Array** (e.g., set to 90° to instantly turn vertical lines into a horizontal meter).
- **Corner Radius & Arc End:** Fine-tune rounded rectangles and the completion angle of arcs.

## Base Size and Anchor

- **Width / Height / Thickness:** The geometric footprint of your shape.
- **Anchor X / Y:** Shifts the pivot point of the shape. If you are building a VU meter needle, shift the Anchor Y downwards so the needle swings from its bottom edge rather than its dead center.

## Repeater (Radial Array)

Enable this to take your single shape and mathematically duplicate it into a perfect circle.

- **Count:** How many copies to generate (e.g., 11 tick marks).

- **Radius:** The distance from the center of the canvas.
- **Rotate Items:** If checked, shapes will point outward (like sunbeams). If unchecked, they will remain perfectly upright (like an orbital ring of LEDs).

## Outer & Inner Shadows

Add realistic depth to your UI components.

- **Outer Shadow:** Casts a shadow outside the shape (creates physical elevation).
- **Inner Shadow:** Casts a shadow inside the shape (creates physical indentations, dimples, or recessed LED tracks).

## Blur

Applies a smooth, hardware-accelerated Gaussian blur to the entire layer. The engine blurs the layer before applying clipping masks.

- **Enable:** Toggles the blur effect on or off.
- **Radius:** Sets the intensity and spread of the blur.

## Texture, Pattern, Grain

- **Pattern:** Hardware-accelerated vector hatching (Dense dots, cross-hatching, diagonal lines).
- **Tile Image:** Overrides the pattern by using your loaded "Image" as a seamlessly tiling texture brush.
- **Grain Intensity:** Injects resolution-independent, procedural film-grain into your shape. Essential for making flat colors look like tactile rubber or matte plastic.

## Gradients

Blend your base color into a secondary color using **Linear**, **Radial**, or **Conical** algorithms. (For rotating LED rings, Conical is your best friend).

## Animation (Start → End)

Every parameter here defines what the layer looks like at Frame 0 (Start) and what it looks like at the Final Frame (End). The engine mathematically interpolates the journey between them.

- **Easing Mode:** How the animation behaves over time.
  - *Linear:* Smooth, constant motion.
  - *Ease-In / Ease-Out:* Accelerates or decelerates.
  - *Stepped (Snap):* Removes smooth motion entirely. The animation "snaps" to hard steps (e.g., set Snap Steps to 3 for a rigid 3-way toggle switch). When applied to

a **Lines Array**, this forces the array to draw in discrete, whole-line chunks (like an LED meter) rather than revealing via a smooth sub-pixel wipe!

- **Snap Steps:** Only active in Stepped mode. Set this to **3** to instantly create a rigid 3-way mechanical toggle switch!
- **Wipe Effect:** Applies an invisible, sliding clipping mask that seamlessly reveals the layer over the course of the animation timeline. If you are building a fader, use a Wipe Effect to reveal the track background, but animate the Offset X/Y on your fader cap layer so it physically moves across the screen to "ride" the leading edge of the wipe!
  - None: The layer renders normally across the entire timeline.
  - Linear Wipes (Up, Down, Left, Right): Reveals the shape directionally. Essential for creating fader tracks, horizontal progress bars, and sliding volume meters.
  - Radial Wipes (CW, CCW): Reveals the shape in a sweeping circle. Essential for circular LED rings, gauge fills, and dial indicators.
- **Animatable Traits:** You can animate Opacity (fade in/out), Angle (rotation), Zoom (scaling), and Offset X/Y (sliding).

## Previewing & Exporting

- **Interactive Scrubber:** Drag the slider to watch your component animate in real-time. What you see is exactly what will be exported.
  - **1:1 Zoom Toggle:** Click the [ 1:1 ] button next to the slider to bypass dynamic viewport scaling. The canvas will instantly snap your component to its exact output pixel dimensions, letting you preview precisely how the UI element will look on your final instrument interface.
- **Export Strip:** Renders your animation into a single, perfectly spaced PNG file ready for your UI framework.
- **@2x:** Automatically renders a second sprite sheet at exactly 200% scale for Retina/4K displays (appends @2x to the filename).
- **Singles:** Exports every single frame as a standalone .png file alongside the main sprite sheet.
- **Horizontal:** Stitches the sprite sheet side-by-side instead of top-to-bottom.
- **Kontakt:** Automatically generates a Native Instruments Kontakt compatible .txt layout file alongside your sprite sheet, pre-configured with your exact frame count and animation orientation.

## GUI Sheet Maker

### Introduction

The GUI Sheet Maker is a dedicated utility for UI/UX designers and plugin developers (especially for platforms like Native Instruments Kontakt). It allows you to take sequences of

individual image frames - such as a rendered 3D knob turning, a slider moving, or a glowing button - and seamlessly stitch them together into a single "filmstrip" or sprite sheet.

Featuring drag-and-drop ordering, live animation auditioning, and flexible grid layouts, this tool eliminates the need for complex image editing software when packaging your UI assets.

## Interface

### Source Images

This section manages the individual frames that will make up your sprite sheet. The order of the images in this list from top to bottom represents the sequence of your animation (from 0% to 100%).

- **Drag & Drop:** You can drag images directly from your computer's file explorer into this list. You can also click and drag items *within* the list to quickly reorder them.
- **Add Images:** Opens a file browser to select multiple `.png`, `.jpg`, or `.bmp` files.
- **Remove Selected:** Deletes the highlighted frames from the list.
- **Move Up / Move Down:** Shifts the selected frame up or down in the sequence.
- **Reverse Order:** Instantly flips the entire list upside down. *Pro Tip: This is incredibly useful if your 3D rendering software exported a knob turning from right-to-left instead of left-to-right.*
- **Clear All:** Empties the list.

### Preview & Settings

#### Preview Mode

- **Full Sprite Sheet:** Displays the fully stitched, static PNG exactly as it will be exported.
- **Animation Frame:** Isolates a single frame and allows you to audition the sequence as a live, moving UI element.
- **Show Gridlines:** When in "Full Sprite Sheet" mode, this toggles a visual cyan overlay showing the boundaries of every cell in your grid. This helps verify that your elements are perfectly centered. *(Note: These gridlines are purely visual and will not be baked into your final exported image).*

#### Sheet Settings

- **Layout:**
  - **Vertical Strip:** Stacks all frames from top to bottom in a single column (the standard format for Kontakt UI elements).
  - **Horizontal Strip:** Places all frames left to right in a single row.
  - **Custom Grid:** Wraps the frames into a 2D grid based on the number of columns you specify.

- **Columns:** Active only in "Custom Grid" mode. Determines how many frames appear in a row before wrapping to the next line (supports massive sheets up to 8192 columns).
- **Pad Mismatched:** Determines how the tool handles images of different sizes.
  - **Checked (Default):** The tool calculates the maximum width and height of all loaded frames, sets the cell size to those maximums, and perfectly centers smaller images inside the transparent cells.
  - **Unchecked:** Forces all images to stretch or squash to exactly match the pixel dimensions of the *first* image in the list.
- **Kontakt:** Automatically generates a Native Instruments Kontakt compatible `.txt` layout file alongside your sprite sheet, pre-configured with your exact frame count and animation orientation.

## Playback & Export

- **Play / Stop:** Only active in "Animation Frame" mode. Starts a live playback of your image sequence.
- **Ping-Pong:** If checked, the animation will loop back and forth (e.g., Frame 1 to 100, then 100 back to 1). This is perfect for auditioning pulsing LEDs or glowing hover states.
- **Speed (ms):** Adjusts the playback speed. Lower numbers equal faster playback (represents milliseconds per frame).
- **Output Size:** A live readout of the final pixel dimensions (Width x Height) of your generated sprite sheet.
- **Save Sprite Sheet:** Compiles the frames and saves the final output as a `.png` file to your computer.

# GUI Sheet Slicer

## Introduction

The Ocean Swift GUI Sheet Slicer is the exact inverse of the GUI Sheet Maker. Where the Sheet Maker stitches a sequence of individual frames into a single sprite sheet, the Sheet Slicer takes a finished sprite sheet and cuts it back into numbered individual frames. It closes the loop in the suite's GUI workflow: both the GUI Knob Maker and the GUI Sheet Maker produce sheets, and this tool is the one that consumes them.

It exists for the situations every UI designer eventually hits - a client or collaborator sends a finished filmstrip with no source frames, a legacy asset needs its frames recovered for re-timing or re-ordering, or a third-party knob strip needs to be broken apart and rebuilt. Sliced frames drop straight back into the GUI Sheet Maker for re-stitching in a different layout.

The cut itself is lossless. Frames are copied out of the sheet by exact pixel coordinates with no resampling of any kind, so a frame that goes into a sheet and comes back out again is bit-for-bit identical to the original.

## Loading a Sheet

Drag a sprite sheet onto the Source Sheet field, or press Browse to select one. PNG, JPG, and BMP files are accepted. Once loaded, the tool reports the file name and the sheet's pixel dimensions.

Very large images are refused before they are opened. The tool reads the image's dimensions from its file header first, and if the sheet exceeds the safety limit it is rejected without ever being decoded into memory — a corrupt or pathological file cannot stall the application.

## Setting the Grid

The Sheet Slicer needs to know how the frames are arranged. There are three ways to tell it, and the first one is usually automatic.

Kontakt sidecar detection: Both the GUI Sheet Maker and the GUI Knob Maker can write a Kontakt .txt metadata file alongside the sheet they export. If such a file sits beside the sheet you load, the Sheet Slicer reads it and configures itself: the frame count and the strip orientation are taken directly from the metadata. For any sheet produced by the suite's own tools with metadata enabled, slicing becomes a two-click operation.

- **Detect:** For sheets with no sidecar, the Detect button makes an informed guess based on the sheet's proportions, assuming square frames. A tall sheet whose height is a whole multiple of its width is read as a vertical strip; a wide sheet whose width is a multiple of its height is read as a horizontal strip. This covers the majority of knob and switch filmstrips. If neither fits, the tool says so rather than guessing wildly, and you set the count by hand.
- **Manual:** Choose the Layout - Vertical Strip, Horizontal Strip, or Custom Grid - and set the number of Frames. For Custom Grid, also set the number of Columns; the row count follows automatically as the frame count divided across those columns, rounded up. This matches the GUI Sheet Maker's own layout logic exactly.

Below the controls, a status line continuously reports the resulting grid and the calculated cell size, and tells you whether the sheet divides evenly.

## The Divisibility Rule

A sprite sheet is a uniform grid: every cell is the same size, with no gaps or margins between frames. It follows that the sheet's width must divide evenly by the number of columns, and its height by the number of rows. If it does not, the frame count or layout is wrong.

When that happens, the status line turns red and shows exactly what does not divide, with the remainder for each axis, and the Slice Frames button refuses to run. This is deliberate. Slicing a mismatched grid would not fail loudly — it would quietly produce a full set of frames, each one

cut through its neighbours, and the damage might not be noticed until much later. Refusing with the numbers on screen tells you immediately what to correct.

## Previewing Before You Commit

Two view modes let you confirm the grid is right before anything is written to disk.

- **Sheet + Grid:** Displays the whole sheet with the slice lines drawn over it as a cyan dashed grid. A wrong frame count is obvious at a glance — the lines will not land on the frame boundaries.
- **Animation Frame:** Plays the sliced frames as an animation, exactly as they would appear in a plugin or Kontakt instrument. Press Play to start, set the Speed in milliseconds, and enable Ping-Pong to bounce the sequence back and forth instead of looping. If the grid does not divide evenly, the preview says so rather than showing a misleading result.

Watching the animation is the most reliable check available: if the frames are cut correctly the motion is smooth, and if the count is off the animation visibly drifts or jumps.

## Auto-Trim

The optional Auto-trim transparent borders checkbox crops fully transparent edges from each sliced frame before saving. It is off by default.

It is worth being clear about what this can and cannot do. When the Sheet Maker builds a sheet from frames of differing sizes, it centres each smaller frame inside a uniform transparent cell — and once that has happened, the original frame size is no longer recorded anywhere. Auto-trim removes transparent margins, which is often what you want, but it is a convenience rather than a reconstruction: it cannot know what the frame's original dimensions were, only where its visible pixels are. A frame that is entirely transparent is left at full cell size rather than being reduced to nothing.

With the option off, every frame comes out at the full cell size, exactly as it sits in the sheet.

## Slicing

Set the Output Folder and the Base name, then press Slice Frames.

Frames are always written as PNG, which preserves transparency and matches what both of the suite's sheet-producing tools use. They are named using the base name followed by a zero-padded three-digit index starting at zero — frame\_000.png, frame\_001.png, and so on — which sorts correctly in any file browser and matches the individual-frame naming used by the GUI Knob Maker. Frames are numbered in reading order: left to right, then top to bottom, the same order the Sheet Maker used to lay them down.

If no output folder is chosen, the frames are written to a folder named after the sheet with "\_frames" appended, created beside the source image. No metadata file is written; the Slicer produces frames, not a sheet, so a sheet sidecar would have nothing to describe.

## Presets

The tool restores its last state on every launch, including the sheet you were working on and your chosen output folder.

Presets store the layout, frame count, columns, auto-trim setting, base name, and the animation preview's speed and ping-pong settings.

- **Save Preset:** Saves the current slicing configuration.
- **Load Preset:** Loads a saved configuration.

# Kontakt NCKP Info Pane

## Introduction

The Kontakt NCKP Info Pane tool is a specialized utility designed for sound designers and instrument developers. It instantly parses Native Instruments `.nckp` (or `.json`) UI files to extract and compile all your Info Pane documentation into a clean, readable list. It is the perfect companion for auditing your instrument's built-in help text or generating external manuals.

## How to Use

- **Load NCKP File:** Click this button to open an OS browser and select your target file. The tool automatically remembers your last-used folder, saving you from navigating your hard drive every time.
- **The Display Area:** Once a file is loaded, the tool recursively crawls the file's architecture. It extracts every valid `infoPaneText` entry and neatly formats it alongside its exact Control ID.
- **Copy to Clipboard:** Instantly copies the entire formatted list to your system clipboard, ready to be pasted into your documentation, spreadsheets, or text editors.

# Kontakt NCKP KSP Persistent

## Introduction

The Kontakt NCKP KSP Persistent tool is designed to automate one of the most tedious aspects of Kontakt instrument development: writing persistence declarations. It scans your Native Instruments `.nckp` (or `.json`) UI files, identifies the actionable controls, and instantly generates a complete list of `make_persistent()` statements ready to be pasted directly into your KSP script.

## How to Use

- **Load NCKP File:** Click to browse your system for a target file. The tool will automatically remember your last-used directory for faster workflow.
- **The Display Area:** Once a file is processed, this area will populate with a clean, formatted list of KSP commands.
- **Copy to Clipboard:** Instantly copies the generated code to your system clipboard, allowing you to paste it straight into your script editor.

## Parsing Options

The tool includes live-updating settings to refine your generated code. Adjusting these while a file is loaded will instantly recalculate the output.

- **Add Prefix (\$):** When checked, the tool automatically prepends the standard KSP variable prefix to your control names (e.g., `make_persistent($my_slider)`). Uncheck this if your specific scripting workflow handles prefixes differently.
- **Ignore Indices (comma separated):** NCKP files assign an "index" number to define the type of UI element. Not all elements need persistence (such as background panels or text labels).
  - By default, this is set to `0, 1` to filter out common static elements.
  - If you notice unwanted items appearing in your list (like an image), you can check its index in the NCKP file and add it to this ignore list to instantly filter it out.

# Kontakt NKS2 Mapper

## Introduction

The Kontakt NKS2 Mapper is a visual authoring environment designed specifically for instrument developers to construct Native Instruments Kontrol MK3 hardware layouts. Instead of

writing dozens of repetitive `set_control_par()` arrays and tracking automation IDs by hand, this tool allows you to visually design your hardware pages and instantly generates the exact, error-free KSP code required for your `on_init` callback.

Built with strict adherence to the NKS2 SDK, the mapper automatically enforces valid Type-to-Style relationships, manages pagination, and intelligently handles complex data structures like `NUM_VALUES` so you can focus on the user experience.

## Groups & Visualizer

The top section of the tool manages the overarching pages of your hardware layout and provides a real-time preview of the physical screens.

- **NKS Page Group:** Select the active group to edit.
  - *Note on Group 0:* Group 0 is strictly locked as the **Performance** group. It is the default view on the hardware, cannot exceed 16 parameters, and cannot be assigned a navigation name.
  - *Edit Groups (1+):* All subsequent groups represent your Edit mode pages. You must assign a **Group Name** to these for them to appear in the hardware navigation menu.
- **Hardware Visualizer:** A 1:1 graphical approximation of an 8-encoder Kontrol MK3 screen. As you change NKS Styles in the configuration table, the visualizer instantly updates to reflect the physical UI (e.g., drawing arcs for knobs, rendering lists for menus, or displaying vector representations of Waveform/Filter shapes). Use the `<` and `>` buttons to page through your parameters if a group exceeds 8 slots.

## The Configuration Table

This is the core workspace where you define the individual parameters for the selected group.

- **Drag & Drop Ordering (≡):** Click and hold the ≡ handle on the far left of any row to drag it up or down. The tool will automatically recalculate the slot indices and update the visualizer in real-time.
- **Section Name:** The overarching category for the knob (e.g., 'Osc 1' or 'Filter'). Adjacent parameters with the exact same Section Name will be visually grouped under a single header line on the hardware.
- **UI Variable:** The exact KSP variable name of your UI control (e.g., `$cutoff_knob`). The engine uses this to automatically fetch the UI ID and bind the automation ID.
- **NKS Type & Style:** Select the behavioral type (Continuous, Discrete, Toggle) and its visual style.
  - *Smart Constraints:* To prevent KSP compiler errors and SDK violations, selecting an NKS Type automatically filters the NKS Style dropdown to only show visually valid pairings.
- **Values / Icons:** A multi-purpose field for defining steps, menus, or graphical icons.

- For Menus, enter a comma-separated list (e.g., `Saw, Sine, Square`). The engine automatically counts the items, sets the mandatory `$CONTROL_PAR_NKS_NUM_VALUES`, and builds the string array loop.
- For Waveforms and Filter Types, hover your mouse over the text box to view a tooltip containing the exact, hardcoded SDK icon strings required by Native Instruments.

## Generated KSP & Presets

The bottom section handles code exportation and session persistence.

- **Generated KSP Editor:** A read-only text field containing your finalized KSP. The code is divided by massive ASCII headers (e.g., `PERFORMANCE GROUP` vs `GROUP 1: FILTER`) to prevent optical illusions and keep your script highly readable. Simply click inside, copy all, and paste directly into your Kontakt script's `on_init` callback.
- **Save / Load Preset:** Save your entire mapping layout—including all groups, parameter orders, and custom strings—to a `.json` file. This allows you to build master templates for complex synths and instantly recall them for future projects.

## Kontakt Performance View

### Introduction

The Kontakt Performance View tool is an advanced, standalone editing environment for Native Instruments `.nckp` GUI files. It provides a visual, drag-and-drop real-time workflow for constructing complex Kontakt interfaces without writing a single line of layout code. Connect to Kontakt directly and visualize your instrument, as you are building it.

Featuring a fully integrated WYSIWYG (What You See Is What You Get) canvas, a dynamic property inspector, and a 50-step Undo/Redo engine, this tool bridges the gap between technical JSON editing and intuitive visual design.

### Global Workflow & Top Controls

The top navigation bar manages your file operations, automatic safety features, and global component insertions.

- **File Menu:** Standard operations for creating **New** files, **Loading**, **Saving**, and **Saving As**. The **Recent** menu automatically tracks your last 10 projects for instant recall.
- **Auto Save:** When enabled, the tool silently saves your project to disk 1.5 seconds after your last edit. This prevents data loss during rapid design sessions.

- When connected directly to Kontakt, enabling auto save also allows seeing editing updates live directly in the instrument UI in Kontakt itself.
- **Auto Load:** When enabled, the suite will automatically load your most recently opened project the moment you launch the Kontakt Performance View tool.
- **Control (Insert):** A dropdown menu containing every valid Native Instruments UI control type (Panel, Button, Slider, Wavetable Visualizer, Level Meter, etc.). Selecting an item injects it into your current selection.
- **Prefab:** A dropdown menu that recursively scans your **Resources/Prefabs** folder for saved **.nckc** (Control) files. This allows you to instantly drop pre-configured, complex UI elements into your project.
- **Copy Prefab Assets:** A utility button that automatically copies all factory prefab images (required for the Ocean Swift factory **.nckc** prefabs) directly to your active instrument's local **pictures** folder.
- **Prefix:** When checked, the tool automatically prepends the correct KSP variable symbol (\$, %, ?, @) to the Variable Name output based on the control type (e.g., % for Tables, ? for XY Pads).
- **View (Eye Icon):** Toggles the interactive WYSIWYG Preview Window.

## The NCKP Tree (Left Pane)

The left pane displays the exact structural hierarchy of your instrument's UI. This custom tree supports drag-and-drop, multi-selection, and intelligent ID collision management.

- **Drag & Drop:** You can drag **.nckp** or **.json** project files directly from your computer onto the tool to load them. You can also drag **.nckc** component files directly onto specific panels in the tree to import them.
- **Renaming:** Double-click any control ID in the tree to rename it. The engine will automatically prevent duplicate names by appending numbers to collisions.
- **Advanced Selection:** **Alt + Click** a Panel to instantly select the Panel and every child control nested inside of it.

## Context Menu (Right-Click)

Right-clicking an item in the tree reveals a powerful context menu:

- **Insert...:** Add a new standard control as a child of the selected item.
- **Select Similar...:** Instantly select all controls of the exact same type either globally across the entire project, or strictly within the current Panel.
- **Cut / Copy / Paste / Duplicate / Delete:** Standard clipboard operations.
- **Rename:** Highlights the ID for editing.
- **Move Up / Move Down:** Shifts the control's rendering order within its parent panel.
- **Import / Export:** Export a single control (and all its children) as an **.nckc** prefab to share or use in other projects, or import a saved prefab.

## Keyboard Shortcuts

- **Undo / Redo:** `Ctrl+Z` / `Ctrl+Y`
- **Cut / Copy / Paste:** `Ctrl+X` / `Ctrl+C` / `Ctrl+V`
- **Duplicate:** `Ctrl+D`
- **Delete:** `Delete` or `Backspace`
- **Move Up / Down:** `Ctrl+Up` / `Ctrl+Down`
- **Import / Export NCKC:** `Ctrl+I` / `Ctrl+E`

## The Property Inspector (Right Pane)

When you select a control in the tree or the canvas, its JSON properties are dynamically flattened and displayed in the Right Pane.

- **Type-Safe Editing:** The table utilizes custom widgets for specific data types. Colors feature a clickable hex-swatch to open a color picker; booleans feature checkboxes; and specific enumerations (Fonts, XY Modes, Alignments, etc.) feature clean dropdown menus.
- **Dynamic Arrays:** If you change a `.length` property (such as the number of entries in a dropdown menu or cursors in an XY Pad), the tool will automatically generate or delete the required JSON blocks behind the scenes to keep your file perfectly structured.
- **Variable Name Footer:** Located at the very bottom of the inspector, this read-only field displays the exact KSP-ready variable name for the selected control, calculating its parent-hierarchy path. Click the **Copy** button to instantly send it to your clipboard.

## The Preview Window (WYSIWYG Canvas)

Clicking the **View** button opens a floating, hardware-accelerated preview canvas. This window renders a close to 1:1 visual representation of your Kontakt UI.

### Canvas Navigation Bar

- **Grid:** Overlays a visual alignment grid (ranging from 10x10 up to 100x100 pixels).
- **Snap:** When enabled, dragging or resizing controls will mathematically snap them to the nearest grid intersection.
- **Play (Test Mode):** Toggles between Edit Mode and Play Mode (detailed below).
- **Values:** When in Play Mode, this displays a dynamic tooltip showing the exact numerical value of the control you are interacting with.
- **Info:** Toggles an information footer at the bottom of the window to display the `infoPaneText` of hovered controls.

## Edit Mode

When Play Mode is *unchecked*, the canvas acts as a layout editor:

- **Selection & Movement:** Click any control to select it (syncing the tree and inspector). Click and drag to physically move the control across the canvas. You can also use the **Arrow Keys** on your keyboard to gently nudge selected items pixel by pixel.
- **Resizing:** When a single valid control is selected, grab handles will appear on its edges and corners, allowing you to freely scale its width and height. (*Note: Certain controls like Panels, basic Knobs, and standard Sliders are restricted from free-resizing to adhere to Kontakt's engine rules*).
- **Alignment Tools:** Select two or more controls, then right-click the canvas to access the Alignment Menu. You can instantly align multiple elements to the Left, Right, Top, Bottom, or perfectly center them horizontally or vertically based on their collective bounding box.

## Play Mode

When Play Mode is *checked*, the canvas transforms into a live synthesizer interface. You can interact with your design exactly as an end-user would, **without** altering your saved JSON coordinates or defaults:

- Drag sliders, turn knobs, and click buttons to watch their sprite-sheet animations update in real-time.
- Move cursors around XY Pads, and click through Table arrays.
- Hold **Ctrl** and click a control to instantly reset it to its default test value.
- Hold **Shift** while dragging a knob or slider to engage fine-tune (slower) movement.

# MIDI 8 Controls

## Introduction

This utility provides a clean, 8 channel mixer interface for sending Continuous Controller (CC) messages to your external hardware synthesizers, virtual instruments, or DAW. It is perfect for testing synth parameters, building custom macro controllers, or generating random patch variations on the fly.

The MIDI 8 Controls tool automatically transmits data to the MIDI Output Port defined in your main application settings.

## Global Actions

These buttons affect all 8 channels simultaneously.

- **Touch (Send All Active):** Instantly broadcasts the current slider values of all active channels (any channel where the CC is not set to -1). This is highly useful for "syncing" your hardware synthesizer to the exact state shown on your screen.
- **Randomize All:** Instantly generates a random value (0-127) for all active channels, updates the sliders visually, and broadcasts the new values to your synth. This is fantastic for generating happy accidents or chaotic patch variations!

## The Mixer Channels

The main interface consists of 8 identical control columns. You can configure each column independently to target different parameters or even different synthesizers.

From top to bottom, each channel features:

- **Custom Name:** Click on the text box (e.g., "Knob 1") to type a custom name for this control, such as "Filter Cutoff" or "Reverb Decay".
- **Value Box:** A precise numerical readout of the current value (0-127). You can type an exact number here and hit Enter, or use the up/down arrows.
- **Vertical Slider:** Click and drag this slider to send smooth, real-time MIDI CC sweeps to your destination.
- **CC Selector:** Sets the MIDI Continuous Controller number (0 to 127) this channel will transmit.
  - *Note: To completely disable a channel so it ignores "Touch" and "Randomize" commands, set this value to -1.*
- **MIDI Channel:** Sets the specific MIDI Channel (1-16) this slider will transmit on. You can have Slider 1 controlling a synth on Channel 1, and Slider 2 controlling a drum machine on Channel 10!

## Utilities

- **Save Preset:** Opens a dialog to save your entire console state to a .json preset file. This saves your custom names, CC assignments, MIDI channels, and current slider values.
- **Load Preset:** Instantly recalls a previously saved .json console configuration. Loading a preset will automatically trigger a "Touch" command to instantly sync your hardware to the loaded values.

# MIDI Generator Nord Drum

## Introduction

The MIDI Generator Nord Drum is a powerful, algorithmic MIDI sequencing tool designed to instantly create massive libraries of royalty-free, hardware-ready drum loops. While specifically mapped and optimized for the unique multi-channel architecture of the Clavia Nord Drum series,

the exported MIDI files can easily be mapped to any drum machine, sampler, or virtual instrument in your DAW.

Rather than relying on static pre-recorded MIDI packs, this tool uses generative blueprints to craft unique patterns across 10 distinct electronic music styles, complete with humanized timing, swing, and performative flourishes like ratchets and ghost notes.

## Output Settings

This section dictates where your generated MIDI libraries will be saved.

- **Base Folder:** Choose the primary destination for the exported files. By default, this points to your Suite's Documents folder.
- **Create timestamped parent folder:** When enabled, every click of the "Generate" button creates a new, uniquely dated folder (e.g., `NordDrumMIDIGenerator_20260302_143000`). This ensures you never accidentally overwrite a previous batch of loops.

## Styles Selection

Select which musical genres the generator should use as a blueprint.

- **Available Styles:** Techno, Trance, Tribal, Glitch, Experimental, Hip Hop, IDM, Dub Techno, Lo-Fi House, and Sparse.
- **Select All / Deselect All:** Quickly toggle your selections. The generator will create dedicated subfolders for each selected style during the export process.

## Generator Parameters

This is the core engine where you define the length, groove, and behavior of the generated MIDI.

- **Bars:** The length of each generated MIDI loop (in 4/4 time).
- **Patterns / style:** The batch size. If you set this to 50 and select 4 styles, the tool will instantly generate 200 unique MIDI files.
- **BPM & PPQ:** Sets the tempo and the MIDI resolution (Pulses Per Quarter note) written into the file's metadata.
- **Swing:** Applies a rhythmic shuffle to the 16th notes. `0.50` is a mathematically straight grid. Values above `0.50` push the off-beats later in time for a classic MPC-style groove.
- **Humanize (ms):** Introduces microscopic, randomized timing imperfections (in milliseconds) to every drum hit, simulating a live drummer.
- **C1 Mapping:** Crucial for Nord Drum users. Select whether your hardware is mapped to receive its base channel starting at MIDI Note 24 or MIDI Note 36.

- **Strict Grid:** A global override. Checking this disables all Swing, Humanization, Ratchets, and Ghost notes, locking the output to a rigid, quantized grid.
- **Seed Settings: \* Random seed each run:** Ensures every single generated file is mathematically unique.
  - **Manual Seed:** Disabling the randomizer allows you to input a specific numerical seed. Using the same seed with the same settings will repeatedly generate the exact same MIDI patterns.
- **Performative Flourishes:**
  - **Ghost hats:** Injects quiet, syncopated hi-hat hits between the main groove.
  - **Ratchets:** Adds rapid-fire, burst-roll subdivisions (commonly used in IDM and Trap).
  - **Kick grace:** Adds subtle, quiet kick drum leading-notes right before a main downbeat.

## Presets & Generation

- **Preset Load / Save:** Save your exact configuration (including style selections and swing parameters) as a custom `.json` preset to recall specific "groove templates" later.
- **Generate / Cancel:** Click Generate to begin the batch process. The tool utilizes a multi-threaded background engine, meaning the Suite will remain responsive and you can safely hit **Cancel** at any time to abort a massive batch render.

## Under the Hood: Technical Details

- **Algorithmic Density:** The generator doesn't just place random notes; it understands musical phrasing. Fills naturally occur at the end of 8, 16, or 32-bar phrases, and element density (like hi-hat complexity) evolves mathematically throughout the loop.
- **State Persistence:** Your exact UI configuration is saved automatically to your Suite documents folder when the tool is unloaded, ensuring your workflow is preserved.
- **Velocity Dynamics:** Every hit is velocity-mapped. Kicks and snares land with specific accent velocities, while flourishes (like ghost notes) are written with significantly lower MIDI velocity values to ensure a dynamic, mix-ready performance out of the box.

## MIDI Keys

### Introduction

The MIDI Keys tool is a responsive, built-in virtual keyboard designed for rapid sound design auditioning. This tool automatically routes its MIDI data to the Global MIDI Output Port selected in the Sound Creator Suite's main application preferences.

## Interface

### Keyboard Settings

These controls determine how your mouse clicks are translated into MIDI data.

- **Channel Dropdown:** Selects the MIDI channel that the notes will be transmitted on.
- **Center Octave Shift:** Shifts the base octave of the keyboard up or down.
  - At 0, the keyboard spans from C2 to C5.
  - Use -1 / -2 for sub-bass auditioning, or +1 / +2 for high-register leads and plucks.
- **Fixed Velocity Slider (1–127):** Because a mouse click lacks physical dynamics, this slider dictates the exact velocity value sent with every Note On message.
- **Octaves:** Determines the total number of visible octaves displayed on the virtual keyboard.
- **Size:** Adjusts the vertical height of the piano keys. This allows you to scale the visual footprint of the keyboard to best fit your current screen layout and workflow.

### The Virtual Keyboard

- **Click to Play:** Left-click any key to send a Note On message. Releasing the mouse sends a Note Off.
- **Glissando (Drag) Support:** Click and hold your left mouse button, then drag your cursor smoothly across the keys to trigger rapid glissandos and sweeps.

### Utility Controls

- **Panic (All Notes Off):** If a rogue note ever gets stuck (e.g., if you are tweaking a synth's release envelope while playing), click this red button. It instantly blasts an "All Notes Off" command across all 128 keys of your currently selected MIDI channel, cutting all sound.
- **Save / Load Preset:** Save your current Channel, Octave Shift, and Velocity settings to a file.

## MIDI Monitor

### Introduction

This utility is designed to help you inspect, troubleshoot, and analyze the MIDI data flowing into the Sound Creator Suite in real-time. Whether you are debugging hardware controllers, verifying synth output, or analyzing complex RPN/NRPN streams, the MIDI Monitor provides a clean, filterable view of your data.

The MIDI Monitor automatically listens to the MIDI Input Port defined in your main application settings.

### Options

Use the top bar to define exactly *how* the data is displayed and *where* it is coming from.

- **Listen On (Channel):** Use this dropdown to filter incoming messages by a specific MIDI channel (1-16), or leave it on **All Channels** to see everything.
- **Verbose Mode:** \* **Checked (Default):** Formats the output into clean, single-line sentences (e.g., `Controller: 1 Value: 64 Channel: 1`). This is highly recommended for reading fast, continuous streams of data.
  - **Unchecked:** Breaks messages into multi-line blocks, mimicking classic hardware displays.
- **Show Channel:** Appends the originating MIDI channel to the end of every message. Turn this off if you are only working on a single channel and want a cleaner display.

## Event Filters

The Event Filter section allows you to declutter your display by ignoring MIDI messages you don't care about. Only checked message types will appear in the monitor.

- **Note On:** Displays incoming MIDI Note On messages, translated to their musical pitch (e.g., `C3 (60)`).
- **Velocity:** Displays the strike velocity (1-127) of Note On messages.
- **Note Off:** Displays incoming Note Off messages.
- **Release Velocity:** Displays the release velocity of a Note Off. *(Note: Many keyboards send a Note On with a velocity of 0 instead of a true Note Off. The monitor smartly catches this and will display <n/a> for the release velocity in those cases).*
- **Controller:** Displays standard Continuous Controller (CC) messages along with their standard MIDI assignment names (e.g., `Controller: 1 (Modulation Wheel MSB)`).
- **Pitch Bend:** Displays high-resolution Pitch Bend values (-8192 to 8191).
- **Channel Pressure:** Displays standard Monophonic Aftertouch values.
- **Poly Aftertouch:** Displays Polyphonic Aftertouch values tied to specific keys.
- **Program Change:** Displays patch/program changes (numbered 1 to 128).
- **RPN / NRPN:** This is a powerful smart-filter. When checked, the monitor will automatically intercept CCs 98, 99, 100, 101, and Data Entry (CC 6). Instead of spamming your screen with raw CCs, it waits for the data entry and cleanly formats them into parsed **Registered Parameter Number (RPN)** or **Non-Registered Parameter Number (NRPN)** messages, identifying common parameters like *Pitch Bend Sensitivity* or *Fine Tuning*.

## Utilities

- **Clear:** Instantly wipes the text display clean. Useful when you want to isolate a specific incoming message without scrolling.
- **Save Preset:** Opens a dialog to save your current combination of Event Filters, Channel routing, and display options to a `.json` preset file.
- **Load Preset:** Instantly recalls a previously saved `.json` configuration.

# MIDI XY Bowl

## Introduction

The Ocean Swift MIDI XY Bowl is a two-dimensional performance surface that emits MIDI on six lanes at once. At its heart is a ball rolling in a bowl: you place it, drag it, or fling it across an XY pad, and its position continuously drives six MIDI CC lanes — A, B, C, D, X, and Y. The four corner lanes (A, B, C, D) are bilinear corner weights that always sum to a full-scale total, forming a quad-fade — where the ball sits decides how much of each corner is active, ideal for morphing between four sounds or parameters by moving through the bowl. The X and Y lanes carry the raw axis positions, giving you two independent macros from the same gesture.

The tool brings the XY Controller module from the Ocean Swift modular environment into the Sound Creator Suite, and adds a physics engine so the ball can have weight and momentum. In Direct mode the ball is simply your pointer. In Physics mode the ball is a mass on a spring inside a bowl with walls: flick it and it arcs, bounces, and settles — or, with friction at zero, orbits forever — painting evolving MIDI across all six lanes from a single throw. It is built in the same spirit as MIDI 8 Controls, sharing that tool's proven MIDI output engine, but the values come from a living physics toy rather than eight faders.

## The Bowl

The large pad is the bowl. Its four corners are labeled A (top-left), B (top-right), C (bottom-left), and D (bottom-right). A green ball shows the current position, with a fading trail behind it so you can see its recent motion. Moving the ball is the core interaction; everything the tool sends is derived from where the ball is.

## Mode

Two modes determine how the ball responds:

- **Direct:** The ball is your pointer. Drag anywhere on the pad and the ball goes there instantly; the six lanes track it exactly. This is the classic XY-pad behavior.
- **Physics:** The ball becomes a mass on a damped spring, tethered to the point you drag to, bouncing off the bowl's elastic walls. Dragging sets the point the ball springs toward; releasing a drag with motion flings the ball. Thrown motion is the output, so a single flick can generate a long, evolving sweep across all six lanes.

Switching modes re-centers the simulation on the current target, so there is never a jump.

## Weight and Friction

These two controls shape the physics in Physics mode:

- **Weight:** Sets how heavy the ball feels by mapping to the spring rate. Light weight is a snappy, fast tracker that follows your drag closely; heavy weight is a slow, loose tether that lets the ball fly almost freely between bounces.
- **Friction:** A single control governing both the damping and how much energy the walls absorb on each bounce. At full friction the ball comes to a dead stop quickly, with fully absorbing walls. As you lower it, motion rings longer and bounces harder. At exactly zero, friction is genuinely lossless: no damping, perfect bounces, and a flick keeps the ball moving forever. The motion is numerically stable, so a zero-friction orbit will not drift or blow up — endless movement is the intended behavior there.

When friction is above zero, a settled ball snaps exactly to rest once it is home and still, so a resting ball is perfectly quiet.

## Driving the Bowl

Beyond dragging and flinging by hand, the tool offers built-in motion sources so the bowl can move on its own:

- **Physics self-motion:** In Physics mode with low friction, one flick keeps the ball orbiting and bouncing indefinitely — a hands-off modulation source in itself.
- **Tether LFOs:** Two free-running LFOs, one for the X axis and one for the Y axis, steer the point the ball chases (in Physics mode) or the position directly (in Direct mode). Each LFO has a Rate in Hz, a Depth, and a Shape (Sine, Triangle, Saw, or Random, where Random holds a new value once per cycle). Two LFOs at different rates trace the classic evolving XY-pad figure. The LFOs are fully free-running — this tool has no tempo sync, by design.

The LFOs steer the tether; the physics still applies on top, so with the ball heavy and friction low the LFO motion becomes a gentle, springy wander rather than a rigid path.

## MIDI Lanes

The six lanes are fixed: A, B, C, D, X, and Y. Each lane row shows its live value (0–127) and offers two settings:

- **CC:** The MIDI CC number this lane transmits on. Set it to -1 to disable that lane completely.
- **Ch:** The MIDI channel (1–16) this lane transmits on.

By default the lanes use CC numbers 20 through 25. The bowl's position is sampled continuously and converted to 0–127 per lane; a lane only sends when its value actually changes, so a still ball produces no MIDI traffic.

The MIDI output port is not chosen inside this tool. Like MIDI 8 Controls, it follows the application-wide MIDI output port set in the app's Settings, and the tool picks up changes to that setting automatically while it is open.

## Touch and Randomize

- **Touch:** Broadcasts the current value of every active lane at once (CC not set to -1), so a receiving device or DAW can be synced or MIDI-learned to the current state on demand.
- **Randomize:** Bowl-native — rather than randomizing fader values, it moves the bowl. In Direct mode it jumps the ball to a random position; in Physics mode it flings the ball with a random velocity. Either way the six lanes follow.

## Presets

On first launch a settings file is created, and the tool restores its last state on every launch. Opening the tool never sends MIDI on its own — the startup restore is silent, so nothing is transmitted until you move the ball or press Touch.

Presets store the mode, weight and friction, the ball's target position, the two LFO settings, and each lane's CC number and channel.

- **Save Preset:** Saves the current configuration to a preset file.
- **Load Preset:** Loads a saved configuration. Loading a preset broadcasts the restored values, unlike the silent startup restore.

## MIDI to OSC Sender

### Introduction

The MIDI to OSC Sender is a dedicated routing matrix that bridges your physical MIDI controllers with modern modular environments. It listens to your incoming MIDI control changes (CC) and instantly translates up to 8 specific parameters into Open Sound Control (OSC) messages sent over UDP.

This tool automatically listens to the **Global MIDI Input Port** selected in the Sound Creator Suite's main preferences.

### Interface & Controls

#### Target OSC Server

These settings dictate where your translated messages are being sent.

- **Target IP:** The network address of the software receiving the OSC data. Leave this as `127.0.0.1` if you are sending data to another program on the same computer.

- **Port:** The UDP port the receiving software is listening on (default is usually 8000).
- **Base OSC Path:** The root address for your messages (e.g., /midi/cc/). The individual row suffixes will be appended to this base path.
- **Connect:** Click this to lock in your settings, open the OSC bridge, and start translating incoming MIDI data.

## MIDI to OSC Routing Matrix

You can configure up to 8 independent routing channels at once.

- **Active (Checkbox):** Easily enable or disable a specific routing row without deleting its settings.
- **Listen CC:** The physical MIDI Control Change number (0–127) you want this row to listen for (e.g., set this to 1 for your Mod Wheel).
- **OSC Address Suffix:** This string is appended to your Base OSC Path.
  - *Example:* If your Base Path is /oceanswift/ and your Suffix is filter, the final OSC address sent over the network will be /oceanswift/filter.
- **Incoming Value (Monitor):** A read-only green display that shows the live, raw incoming MIDI value (0–127) so you can verify your hardware is mapped correctly.

*Note on Value Translation: To ensure maximum compatibility with modern OSC standards, the raw 0–127 MIDI values are automatically normalized into a float between 0.0 and 1.0 before being sent over the network.*

## Utility Controls

- **Save / Load Preset:** Save your entire routing matrix, target IP, and base path to a file.
- 

# NKS Controller Mapper

## Introduction

The **NKS Controller Mapper** is a powerful batch-processing utility designed for sound designers and preset creators. It allows you to define a single, perfect hardware control layout for a synthesizer by a very simple declarative script and instantly inject that layout across an entire folder of .nksf presets.

Instead of manually turning knobs and saving presets one by one in Maschine or Komplete Kontrol, this tool lets you write your ideal map in plain text and mass-apply it in seconds.

## The User Interface

- **Selected Folder:** Displays the current directory the tool will scan for .nksf files.

- **Select Folder (Button):** Opens an OS dialog to choose the folder containing the presets you want to process. The tool remembers your last used folder.
- **Mapping Field:** A multi-line text box where you write or paste your hardware layout map.
- **Load Map from NKSF (Button):** The easiest way to start. Select any existing `.nksf` file, and the tool will extract its current knob mapping and translate it into plain text inside the Mapping Field.
- **Execute (Button):** Scans the selected folder and safely overwrites the NICA (Native Instrument Controller Assignments) chunk in every `.nksf` file with your new map.
- **Cancel (Button):** Safely stops a batch operation currently in progress.

## Writing a Controller Map (The Syntax)

The tool uses a highly flexible, human-readable text format. You define each knob on its own line using key-value pairs.

### The Basic Format

To map a knob, type the following on a single line:

`Knob = 1, Name = Cutoff, Section = Filter, ID = 18`

### The Rules of the Mapper

The parser is designed to be forgiving and intelligent:

1. **Mandatory Fields (Knob and ID):** Every line *must* have a Knob number and a host automation ID number. If either is missing or isn't a number, the tool will abort the batch to protect your files.
2. **Optional Fields (Name and Section):** If you omit the Name or Section, the tool will safely leave those blank on the hardware screen.
3. **Order Doesn't Matter:** `ID = 5, Knob = 1` works exactly the same as `Knob = 1, ID = 5`.
4. **Spacing Doesn't Matter:** `Knob=1,ID=5` works just as well as `Knob = 1, ID = 5`.

## Advanced Mapping: Pages and Blank Spaces

Native Instruments hardware organizes parameters into **Pages of 8 Knobs**. The Mapper tool automatically calculates your pages and handles blank spaces for you.

### Creating Multiple Pages

To put a parameter on **Page 2**, simply assign it to Knob 9 or higher.

- `Knob = 1` → Page 1, Knob 1
- `Knob = 8` → Page 1, Knob 8
- `Knob = 9` → Page 2, Knob 1
- `Knob = 17` → Page 3, Knob 1

## Leaving Blank Spaces (Skipping Knobs)

You do not need to define every single knob. If you want a visual gap between your Filter controls and your Envelope controls, simply skip a number.

### Example:

Knob = 1, Name = Cutoff, Section = Filter, ID = 4

Knob = 2, Name = Resonance, Section = Filter, ID = 5

Knob = 5, Name = Attack, Section = Amp Env, ID = 12

Knob = 6, Name = Decay, Section = Amp Env, ID = 13

*In this example, the tool will automatically hide Knobs 3, 4, 7, and 8 so they appear blank on your Complete Kontrol keyboard.*

# NKS Create Preset

## Introduction

The **NKS Preset Creator** is an automation utility designed to batch-export presets from software synthesizers hosted in Maschine as NKS presets.

Because most plugins do not allow you to mass-export their internal presets into the **.nksf** format natively, this tool acts as a "virtual assistant." It systematically sends a command to change the preset, executes a custom sequence of mouse macros and keyboard shortcuts to trigger the host's "Save As" dialog, and repeats the process until the entire bank is exported.

## Interface Overview

### Patch Source Settings

This section tells the tool what presets it is trying to export.

- **Folder Button:** Click this to select the specific folder on your hard drive containing your synthesizer's native patch files.
- **Extension:** Enter the exact file extension of your synthesizer's patches to filter the folder (e.g., **.sbf** for Spire, **.h2p** for u-he, etc.).
- **Program Start / End:** Define the starting and ending index within the bank that you want to export. This ensures you don't have to run an entire folder if you only need to process a specific range.

### Navigation Settings (Nav Mode)

This dictates how the script advances to the next preset in your synthesizer - whether to use MIDI Program Changes or simulate clicking the Plugin's 'Next' arrow.

- **Nav Mode Dropdown:** Select either **MIDI PC** or **GUI Next**.
- **MIDI Program Change Mode:** Navigates presets by sending MIDI PC messages.
  - **PC Offset:** A specialized control for synths that start at Patch 1 instead of 0. Leave the PC Offset at 0 for standard synths.
- **GUI 'Next' Button Mode:** Navigates presets by simulating a mouse click on the plugin's internal "Next Preset" button. This is crucial for synths that suffer from MIDI PC desync. *(Note: If using this mode, you must provide the X/Y coordinates for the plugin's "Next" button).*

### Maschine Macro Settings (GUI Automation)

Because the "Save As..." button can be anywhere depending on your monitor resolution and Maschine window size, you must provide the exact screen coordinates for the tool to click.

- **Channel Menu X/Y:** Enter the X and Y screen coordinates for the Maschine Channel Menu dropdown arrow.
- **Save As Menu X/Y:** Enter the X and Y screen coordinates for the "Save As..." item that appears after the Channel Menu is clicked.

### Timing Settings (Crucial for Stability)

Automation scripts move faster than DAWs can process. These wait times prevent the script from clicking before Maschine is ready.

- **Plugin Load Wait:** The time (in seconds) to wait *after* sending the Program Change. This gives the plugin enough time to load the patch's audio and UI before the script attempts to open the Maschine menu. Increase this for CPU-heavy patches.
- **Post-Save Wait:** The time (in seconds) to wait *after* pressing 'Enter' to confirm the save. This gives Maschine the necessary time to finish writing the **.nksf** file to your hard drive before the script sends the next MIDI Program Change.

## NKS Preset Tagging

### Introduction

Welcome to the NKS Preset Tagging Tool. It is designed to automate and manage the incredibly tedious process of tagging Native Instruments NKS preset files (**.nksf**).

Instead of manually clicking through hundreds of patches in the Komplete Kontrol or Maschine browsers, this tool uses a sophisticated Lexical Engine to analyze your presets and automatically inject the correct standard NKS Types (e.g., Bass , Soundscapes ) and Characters (e.g., Analog , Tempo-synced , Dirty ).

Currently, the tool is only compatible with `.nksf` format NKS files.

## The Interface

The tool operates using a classic 3-pane IDE layout for maximum speed and visibility:

- **Left Pane (Smart Filters):** A dynamic accordion tree that scans your loaded folder and lets you instantly filter by Products, Banks, Sub-Banks, Types, and Characters.
- **Middle Pane (Result Browser):** Displays the `.nksf` files matching your filters. It supports text search and multi-selection. Any preset with unsaved edits is distinctly marked with an asterisk (\*).
- **Right Pane (Inspector):** The control center for the presets currently selected in the browser, divided into **TYPES**, **CHARACTERS**, and **PROPERTIES** tabs. It features tri-state checkboxes for bulk non-destructive editing.
- **Audio Previews:** Located beneath the Result Browser, checking "Auto-Play Previews" will instantly audition `.nksf.ogg` files found in the `.previews` subfolder when a single preset is selected.

## How It Works: The Lexical Engine (Text Parsing)

The Lexical Engine acts like an expert sound designer reading a patch list. It analyzes the file name and the parent folder structure to intelligently deduce what the sound is and derive relevant tags from the official NKS taxonomy.

- **Folder Inheritance:** If you place a preset named `Dark Matter.nksf` inside a folder named `Soundscapes`, the engine automatically tags it as a Soundscape.
- **Plugin Features Inheritance:** The engine infers basic tags deriving from the plugin the preset belongs to. For example, every preset for a dedicated granular instrument will automatically mark the "Granular" , "Synthetic" , "Digital" , and "Sample-based" characters.
- **Smart Abbreviations:** It recognizes standard workstation shorthands (e.g., LD = Synth Lead , PD = Synth Pad , BS = Bass ).
- **Contextual Characters:** It scans for descriptive adjectives. Words like "Filthy" trigger the Dirty character , "Retro" triggers Lo-Fi , and "Space" triggers Wet.
- **Conflict Resolution:** It is smart enough to know that a "Hard Sync" lead is not necessarily a "Distorted" audio signal, and that "12-String" refers to an Acoustic Guitar , not a Bowed Strings section.

- **Smart Keyword Parsing:** The engine safely and accurately captures descriptive words and multi-word phrases. It uses precise matching rules—including whole-word isolation, exact phrases, prefixes, and suffixes—to guarantee accurate tagging while strictly eliminating false positives (e.g., successfully tagging Hi-Hat when it sees "Hat", but safely ignoring the word "That").
- **Deterministic Inference:** The engine thinks holistically. By analyzing combinations of tags it initially finds, it can logically deduce and apply secondary traits. For example, if it flags a sound as both Monophonic and Synthetic (and it isn't a bass or drum), it automatically infers that it belongs in the Synth Lead > Classic Mono category.

## Auto Tag Controls

Located in the top bar, the Auto Tag system allows you to unleash the Lexical Engine on any selected group of presets in the Result Browser.

- **Types:** Allows the tool to apply instrument category type tags (e.g., Piano / Keys , Mallet Instruments ).
- **Chars:** Allows the tool to apply the descriptive character tags (e.g., Monophonic , Evolving , Acoustic ).
- **Preserve:** Merges existing tags inside the preset with the new tags found by the engine. The imported tags also take part in the deterministic inference.
- **Acoustic as Sample:** Bypasses engine logic to aggressively tag physical/acoustic instruments as Sample-based. Useful for ROMplers or Kontakt libraries.

## Properties Tab (Metadata Writer)

Located in the Right Pane Inspector, the Properties tab allows you to bulk-edit standard metadata across all selected presets simultaneously. If a field displays **[Multiple Values]**, it means the selected presets currently have conflicting data. Typing a new value will overwrite the conflict for all selected files.

- **Product:** The name of your synthesizer or library (e.g., Oscillarys).
- **Bank:** The primary expansion or factory bank name.
- **Sub-Bank:** A secondary categorization (e.g., Artist Presets).
- **Vendor:** Your company name (e.g., Ocean Swift Synthesis).
- **Author:** The sound designer's name.
- **Comment:** General release notes or website URLs.

## Committing and Clearing Changes

Because the Inspector is entirely non-destructive, checking a box or running the Auto Tag engine only updates the tool's internal memory (marking the preset with an \*).

- **Clear Changes:** Instantly drops all unsaved modifications across all loaded presets, reverting them back to their exact state on your hard drive.
- **Commit Changes:** Safely writes all modified tags, characters, and properties directly into the `.nksf` files, clearing the dirty states.

# NKS Sample Tagging

## Introduction

Welcome to the NKS Sample Tagging Tool. It is designed to automate and manage the incredibly tedious process of tagging standard audio files (`.wav`) with Native Instruments NKS metadata.

Instead of manually clicking through hundreds of individual samples in the Maschine or Komplete Kontrol user browsers, this tool allows you to rapidly bulk-edit your libraries, inject standard NKS Types (e.g., **Drums**, **Percussion**) and Characters (e.g., **Acoustic**, **Sample-based**), and visually inspect audio files for loop points.

Currently, the tool is only compatible with `.wav` format audio files.

## The Interface

The tool operates using a classic 3-pane IDE layout for maximum speed and visibility:

**Left Pane (Smart Filters):** A dynamic accordion tree that scans your loaded folder and lets you instantly filter your samples by Content Types (Loops or One-shots), Products, Banks, Sub-Banks, Types, and Characters.

**Middle Pane (Result Browser):** Displays the `.wav` files matching your filters. It supports text search and multi-selection. Any sample with unsaved edits is distinctly marked with an asterisk (\*). You can also right-click any item and select "Reveal in file system" to instantly locate it on your OS.

**Right Pane (Inspector):** The control center for the samples currently selected in the browser, divided into **TYPES**, **CHARS**, and **PROPS** tabs. It features tri-state checkboxes for bulk non-destructive editing.

**Audio Previews & Waveform:** Located beneath the Result Browser, checking "Auto-Play" will instantly audition the `.wav` file natively when a single sample is selected. You can also manually trigger playback using the **Play** and **Stop** buttons. The custom visualizer renders the amplitude envelope of the selected sample, providing immediate visual feedback of the audio data.

## How It Works: Audio & Loop Parsing

The NKS Sample Tagging tool is built to analyze the binary structure of your audio files.

- **Waveform Extraction:** The tool rapidly scans the binary `data` chunk of the `.wav` file using an optimized stride-read algorithm. This safely extracts an amplitude envelope for the visualizer without loading heavy audio files entirely into memory.
- **Loop Detection:** The tool parses the file's header to locate standard `smp1` chunks. If loop markers were embedded into the file by a DAW or audio editor, the tool reads them.
- **Visual Feedback:** Any detected loop regions are painted directly onto the waveform display as transparent orange shaded areas bounded by dashed markers, letting you instantly verify if a sample is a loop or a one-shot just by clicking it.

## Auto Content Type

Located in the top bar, the Auto Content Type button leverages the internal loop detection to automate your workflow.

Simply select a batch of samples in the Result Browser and click **Auto Content Type**. The tool will instantly iterate through the selection, reading the binary headers. If it finds embedded loop points in a file, it automatically tags it as a **LOOP**. If no loop markers are found, it safely tags it as a **ONESHOT**.

## Properties Tab (Metadata Writer)

Located in the Right Pane Inspector, the **PROPS** tab allows you to bulk-edit standard metadata across all selected samples simultaneously. If a field displays `[Multiple Values]`, it means the selected samples currently have conflicting data. Typing a new value will overwrite the conflict for all selected files.

- **Product:** The main name of your sample pack or library.
- **Bank:** The primary categorization folder.
- **Sub-Bank:** A secondary categorization folder.
- **Vendor:** Your company name (e.g., Ocean Swift Synthesis).
- **Author:** The sound designer's name.
- **Content Type:** A dedicated dropdown to explicitly classify the sample as a Loop or a One-shot.
- **Comment:** General release notes or website URLs.

## Committing and Clearing Changes

Because the Inspector is entirely non-destructive, modifying text, checking a box, or running the Auto Content Type button only updates the tool's internal memory (marking the sample with an `*`).

**Clear Changes:** Instantly drops all unsaved modifications across all loaded samples, reverting them back to their exact metadata state on your hard drive.

**Commit Changes:** Safely writes all modified tags, characters, and properties directly into the `.wav` files. The tool utilizes memory-safe block streaming to inject the new NKS ID3 chunk into the header without altering or degrading the raw audio data, clearing the dirty states once finished.

# OSC 8 Controls User Manual

## Introduction

The OSC 8 Controls tool is a customizable, 8-channel virtual mixer that sends Open Sound Control (OSC) data across your network. Designed to interface effortlessly with modular software like TouchDesigner, Reaktor, or Max/MSP, this tool provides network transmission without causing any stuttering in your main interface.

## Interface & Controls

### Target OSC Server

These settings dictate the network location and formatting for your outgoing OSC data.

- **Target IP:** The network address of the software receiving the OSC data. Use 127.0.0.1 if you are sending data to another program on the same computer.
- **Port:** The UDP port the receiving software is listening on (default is usually 8000).
- **Base OSC Path:** An optional root address for your messages (e.g., `/oceanswift/`). The suffixes defined in the individual channel strips below will be automatically appended to this base path.
- **Connect:** Click this to lock in your settings, initialize the UDP client, and enable the sliders.

### Global Actions

- **Touch (Send All Active):** Instantly captures the current position of all 8 sliders and broadcasts them simultaneously. This is highly useful for initializing your receiving software to match the current state of your suite.
- **Randomize All:** Scrambles the values of all active channels within their individually defined boundaries and immediately transmits the new data. Excellent for generating instant variations when controlling synthesis parameters.

## OSC Controllers (Channel Strips)

Each of the 8 vertical channel strips operates independently.

- **Custom Name:** A purely visual text field allowing you to label what the slider is controlling (e.g., "Filter Cutoff" or "Reverb Decay"). This label is saved with your presets.
- **Maximum / Minimum Boundaries:** Set custom integer limits for the top and bottom of the slider's range (defaulting to 0 and 10000). The slider smoothly interpolates between these two values. You can even invert them (e.g., Min 100, Max 0) for reverse fader action.
- **Value Input & Slider:** Drag the slider or type a precise integer into the box to instantly transmit an OSC message within your defined boundaries. Changing the boundaries will mathematically preserve your current absolute value while visually updating the fader's position.
- **Float Toggle:** When enabled for a specific channel, the current integer value is mathematically normalized between your Min and Max boundaries and transmitted over the network as a high-resolution decimal between 0.0 and 1.0. If unchecked, the raw integer value is sent.
- **OSC Suffix:** This text is combined with your Base OSC Path to create the final address. For example, if your base path is /mix/ and your suffix is fader1, moving the slider will broadcast to /mix/fader1 (the tool automatically prevents dangling slashes and cleans up accidental double-slashes).
- **Active Toggle:** Uncheck this box to completely disable transmission for the specific channel. Disabled channels will ignore "Touch" and "Randomize All" commands.

## Utility Controls

- **Save / Load Preset:** Save your entire mixer state, network target configuration, custom boundaries, float toggles, and custom labels to a file. This allows you to instantly recall specific controller layouts for different projects or live performance setups.

# OSC Monitor

## Introduction

The OSC Monitor is a dedicated, real-time diagnostic tool for viewing incoming Open Sound Control (OSC) messages over your network. It is an essential utility for debugging cross-application routing, modular synthesizer setups, and audio-visual bridges like TouchDesigner or Max/MSP.

Running on a thread-safe background server, it effortlessly captures high-speed UDP network traffic without causing any stuttering in your main interface.

## A Note on OSC Networking

Unlike MIDI, which uses physical or virtual hardware ports, OSC operates over standard network protocols. To capture data successfully, you must ensure that the software sending the OSC messages is pointing to the exact same IP Address and Port that this monitor is listening on.

## Interface & Controls

### Network Connection

These settings determine where the tool looks for incoming network traffic.

- **Listen IP:** The local network interface to bind to.
  - Leave this as `127.0.0.1` to listen exclusively to local traffic (other software running on the exact same computer).
  - Change it to `0.0.0.0` if you want to capture messages being sent from external devices on your local Wi-Fi or Ethernet network (such as an iPad controller).
- **Port:** The specific UDP port to listen on (defaults to `8000`).
- **Listen:** Click this toggle to open the network socket and begin capturing background data.

### Filtering & Display Options

OSC data streams can be overwhelming. These filters help you isolate exactly what you need to see.

- **Address Filter:** If you type a string here (for example, `/cc/1` or `/oceanswift`), the monitor will completely ignore any message that does not contain that text in its address path. Leave this blank to capture all incoming messages.
- **Ignore Duplicates:** OSC hardware often floods the network with redundant data (like a fader continuously broadcasting its resting position). Enabling this prevents the log from flooding by dropping any message that is completely identical to the message immediately preceding it.
- **Show Timestamps:** Prepends the exact local system time (down to the millisecond) to each log entry so you can accurately trace event timing and latency.
- **Show Argument Types:** OSC arguments can be floats, integers, strings, or booleans. Enabling this explicitly prints the data type before the value (e.g., logging `float: 0.5` instead of just `0.5`). This is incredibly useful when debugging why a receiving application is rejecting your data.

### The Log Display

- **Message Log:** A read-only, continuously updating text box displaying your captured and filtered OSC traffic.
- **Clear Log:** Instantly wipes the display clean, giving you a fresh slate for your next test.

## Utility Controls

- **Save / Load Preset:** Save your current IP, Port, and specific filtering configurations to a file.

# OSC XY Bowl

## Introduction

The Ocean Swift OSC XY Bowl is the OSC counterpart to the MIDI XY Bowl, completing a familiar pairing in the suite: just as MIDI 8 Controls has its OSC twin in OSC 8 Controls, the MIDI XY Bowl has its twin here. It is the same two-dimensional performance surface — a ball rolling in a bowl, driving six lanes named A, B, C, D, X, and Y — but instead of MIDI CC it transmits Open Sound Control over the network. The four corner lanes (A, B, C, D) are bilinear corner weights that form a quad-fade, and the X and Y lanes carry the raw axis positions.

The bowl itself — the pad, the ball and its trail, the Direct and Physics modes, the Weight and Friction controls, the two free-running tether LFOs, and the bowl-native Randomize — works exactly as in the MIDI XY Bowl. Please see the MIDI XY Bowl manual section for a full description of the bowl, its physics, and the LFO drivers; that behavior is identical here and is not repeated. This section covers what is specific to the OSC version: connecting to an OSC receiver, the value model, and the per-lane OSC addressing.

### The Target and Connecting

Unlike the MIDI XY Bowl, which uses the application-wide MIDI output port, the OSC XY Bowl connects to a network target from within the tool, in the same way as OSC 8 Controls. A Target OSC Server row at the top holds:

- **Target IP:** The IP address of the software receiving the OSC data (default 127.0.0.1, the local machine).
- **Port:** The UDP port the receiver is listening on (default 8000). **Base OSC Path:** The root OSC address that every lane's suffix is appended to (default /oceanswift/). **Connect:** A toggle button that starts and stops the OSC sender.

When you press Connect, the connection is validated immediately, on the spot, before the button commits — if the IP or hostname cannot be resolved, an error is reported and the tool stays cleanly disconnected rather than failing silently in the background. Once connected, the button reads Disconnect and the IP and Port fields lock. On a successful connection the tool performs an Auto-Sync: it immediately broadcasts the current value of every enabled lane, so the receiver is in step with the bowl from the moment you connect.

The bowl animates at all times, whether connected or not — you can drag, fling, and watch the ball and the lane values move freely. However, OSC is transmitted only while connected. A moving bowl with no connection simply shows its values on screen and sends nothing; there is

no error and nothing is lost. Disconnecting stops transmission and unlocks the target fields so you can point the tool somewhere else. The target can also be changed and reconnected at any time.

## The Value Model

MIDI is inherently limited to 7-bit values (0–127). OSC has no such limit, so the OSC XY Bowl lets you choose how the lane values are sent, with a single global Send as control next to the Touch button:

- **Float 0..1:** Sends each lane's native value directly as a floating-point number between 0.0 and 1.0. This is the natural form for the bowl, whose corner weights and positions are already normalized, and is the recommended choice for most OSC receivers.
- **Integer:** Scales each lane's 0.0–1.0 value to a whole number between 0 and a maximum you set in the Integer max field. The default maximum is 127, which matches the MIDI XY Bowl's resolution, but you can raise it (up to 99999) for finer steps.

Switching the send mode, or changing the integer maximum, updates the on-screen lane values immediately and does not cause a burst of messages — transmission resumes naturally as the bowl moves.

## OSC Lanes

The six lanes are fixed: A, B, C, D, X, and Y. Each lane row shows its live value and offers two settings:

**Suffix:** The text appended to the Base OSC Path to form this lane's full address. The defaults are the lane letters lowercased — a, b, c, d, x, y — so with the default base path the lanes send to /oceanswift/a, /oceanswift/b, and so on. Editing a suffix retargets that lane.

**On:** A checkbox that enables or disables transmission for this lane. Unchecking it silences the lane without losing its suffix — the OSC equivalent of disabling a MIDI lane.

The full address is assembled by joining the base path and the suffix, then cleaned to valid OSC form: a leading slash is ensured, any doubled slashes are collapsed, and a trailing slash on the leaf is removed. For example, a base path of /mix/ and a suffix of ch1 produce /mix/ch1. Each lane sends only when its outgoing value changes, so a still ball produces no network traffic; in Float mode the change is measured to three decimal places.

## Touch and Randomize

- **Touch:** Broadcasts the current value of every enabled lane at once, so the receiver can be re-synced to the current state on demand. Like all transmission, Touch only sends while connected.

- **Randomize:** Bowl-native, exactly as in the MIDI XY Bowl — in Direct mode it jumps the ball to a random position, and in Physics mode it flings the ball with a random velocity. The six lanes follow.

## Presets

On first launch an init.json settings file is created, and the tool restores its last state on every launch. Opening the tool never connects and never transmits on its own — the startup restore is silent, and OSC only flows once you connect.

Presets store the target IP, port, and base path; the mode, weight, and friction; the ball's target position; the send mode and integer maximum; the two LFO settings; and each lane's suffix and enabled state.

- **Save Preset:** Saves the current configuration to a preset file.
- **Load Preset:** Loads a saved configuration. If the tool is connected when you load a preset, it retargets to the preset's server and broadcasts the restored values; the silent startup restore does not.

## OSC to MIDI Server

### Introduction

The OSC to MIDI Server tool is a powerful routing tool that bridges the gap between Wi-Fi-based Open Sound Control (OSC) applications (like TouchOSC or Lemur) and your MIDI hardware or digital audio workstation (DAW). It runs a lightweight background server to listen for incoming network messages and instantly translates them into standard MIDI Control Change (CC) data across 8 independent channels.

### Quick Start Guide

Follow these steps to get your tablet or phone talking to your MIDI gear in under a minute:

1. **Configure MIDI Output:** Before starting, ensure you have selected a valid MIDI Output port in the main Sound Creator Suite settings.
2. **Set Up Your Network:** Ensure your computer and your tablet/phone are connected to the exact same Wi-Fi network.
3. **Start the Server:** In the tool, leave the Local IP as 0.0.0.0 and Port as 8000. Click **Start Server**. *(Note: If Windows Defender or macOS Gatekeeper prompts you, grant the application network access).*

4. **Configure Your Device:** Open your OSC app (e.g., TouchOSC) on your mobile device. Set the Host IP to your computer's local network IP address (e.g., `192.168.1.55`) and the Outgoing Port to `8000`.
5. **Learn a Control:** Click the **Learn** button on Channel 1 in the tool. Move a fader or knob on your tablet. The tool will automatically capture the OSC address, and moving the fader will now send MIDI CC data!

## Interface Breakdown

### Server Settings

This section controls the background network listener.

- **Local IP:** Defaults to `0.0.0.0`. This tells the server to listen for incoming connections on all available network adapters. You generally do not need to change this.
- **Port:** The UDP port the server listens on. Defaults to `8000`. This must exactly match the "Outgoing Port" configured in your mobile OSC app.
- **Start/Stop Server:** Toggles the background listener. You must stop the server before you can edit the IP or Port settings.
- **Status Indicator:** Displays whether the server is currently online (green) or offline (gray).

### The 8-Channel Mixer

The tool provides 8 independent routing strips. Each strip can listen to a specific OSC address and translate it to a specific MIDI destination.

- **Visual Slider:** Provides real-time visual feedback when an OSC message is received. You can also manually click and drag this slider with your mouse to test your MIDI output routing directly from your computer.
- **OSC Addr:** The specific network address the channel is listening for (e.g., `/1/fader1`).
- **Learn Button:** The fastest way to map controls. Click to arm the channel (the button turns red). The very next OSC message received by the server will automatically populate the "OSC Addr" field.
- **MIDI Ch (1-16):** The target MIDI channel for the outgoing message.
- **MIDI CC (0-127):** The specific Continuous Controller number to send (e.g., CC 7 for Volume, CC 74 for Filter Cutoff).

### Presets and Routing Recall

If you use different OSC layouts for different synthesizers or live sets, you can save your entire 8-channel routing matrix.

- **Save Preset:** Exports your current IP, Port, and all 8 Channel Strip configurations to a custom `.json` file on your computer.
- **Load Preset:** Instantly recalls a previously saved routing configuration.
- **Auto-Save:** The tool automatically remembers your exact layout and server settings when you close the Sound Creator Suite and will restore them the next time you open the tool.

## Troubleshooting

- **The server is running, but the sliders aren't moving:** Double-check that your computer and mobile device are on the exact same Wi-Fi network. Ensure your mobile device is sending to the correct computer IP address and that your OS firewall is not blocking the Sound Creator Suite.
- **The sliders are moving, but my synth isn't responding:** Verify that a valid MIDI Output is selected in the main Sound Creator Suite settings menu. Check that the target MIDI Channel and CC number in the tool match what your hardware or DAW is expecting to receive.

## Sampler Auto MIDI

### Introduction

Welcome to the Auto Sampler MIDI. This tool transforms your computer into a robotic sampling assistant. Designed specifically for capturing hardware synthesizers, it automatically sends MIDI Note and Program Change messages to your external gear, captures the audio return, processes it, finds perfect seamless loops, and saves it to disk - all completely unattended.

Whether you want to clone a single bass patch or automatically sample an entire bank of 128 analog pads overnight, this tool handles the heavy lifting so you can focus on sound design.

### File Controls

Define where your newly minted samples will live and how they will be named.

- **Output Folder:** Click **Folder** to select the destination for your recorded `.wav` files.
- **Base Name:** The core name of your sample (e.g., `Deep_Bass`).
- **Appends:** Automatically tag the file name with the exact MIDI data used to create it:
  - *Append Program:* Adds the Program Counter number (e.g., `001`).
  - *Append Note:* Adds the MIDI note triggered (e.g., `Note60`).
  - *Append Velocity:* Adds the MIDI velocity (e.g., `Vel100`).
  - *Append With:* Choose the separator (Space, Underscore, or Dash) to keep your file names clean and readable.
-  **Note on Overwriting:** Because this tool is designed for rapid, iterative batch sampling, **it will automatically overwrite** existing files with the exact same name. This

allows you to tweak a hardware envelope and re-run a sequence without cluttering your hard drive.

## MIDI Controls

Configure the exact MIDI signals sent out to your hardware synthesizer. (*Note: Ensure your MIDI Output Port is correctly selected in the main app settings*).

- **MIDI Channel:** Select the output channel (1-16) your hardware is listening to.
- **Note & Velocity:** The specific pitch (0-127) and velocity (0-127) to trigger.
- **Sustain Time:** How long the engine holds the note before sending the Note-Off message. Options range from 1 second up to 120 seconds for massive evolving pads.
- **Trigger (Button):** A quick-test utility. Click this to instantly fire a 200ms Note-On/Note-Off sequence. Use this to verify your MIDI routing and audio return levels before committing to a long batch recording.

## Program Controls (Batch Sampling)

Automate the process of sampling multiple presets from your hardware in one click.

- **Program Change Checkbox:** When enabled, the tool will automatically step through a range of presets.
- **Bank (LSB):** Selects the specific Bank (0-127) on your hardware synth.
- **Program Start & End:** Define the range of patches to sample (e.g., from patch 0 to patch 10).
- **Load Wait:** Hardware synthesizers often take time to load patches or switch DSP states, which can cause audible pops. This setting tells the engine to pause (0.0s to 5.0s) *after* sending the Program Change but *before* recording the audio, guaranteeing perfectly clean captures.

## Audio & Process Controls

The Auto Sampler MIDI doesn't just record; it applies mastering-grade DSP to format your samples for immediate use in software oscillators.

- **Channels:** Choose to capture a **Stereo** file, or isolate a mono signal from the **Left** or **Right** inputs.
- **Release Threshold:** After the MIDI Note-Off is sent, the engine listens to the audio tail. Once the audio falls below this volume threshold (e.g., -60dB) and stays there for 1 full second, the recording safely stops.
- **DC Kill:** Automatically removes any DC offset or zero-bias from the recorded audio, preventing clicks and maximizing headroom.
- **Normalize:** Peak-normalizes the recorded audio to -1.0dB. *Note: This is applied per channel to ensure maximum volume.*

- **Find Loop (MIR Engine):** When checked, the tool runs a complex Music Information Retrieval (MIR) algorithm. It analyzes the perceived timbre (MFCCs) of the sample's sustain phase to find a perfectly matching, click-free loop of at least 500ms.
  - **Crossfade:** Applies a 10ms crossfade between the loop end and start points.
  - **Crop Loop:** Deletes all audio outside of the detected loop, exporting only the looping section.

## Recording Controls

Set the final export quality of your sampled **.wav** files.

- **Bit Depth:** Choose between 16-bit (int), 24-bit (int), or 32-bit (float).
- **Sample Rate:** Select your target sample rate, from 44100 Hz up to 96000 Hz. *If this differs from your interface's current running sample rate, the engine will automatically invoke high-quality resampling during the save phase.*

## Main Actions & Presets

- **Record / Cancel:** Starts the automated MIDI sequence. If you realize you made a mistake mid-batch, clicking **Cancel** will instantly intercept the sequence, send a safety Note-Off to silence your hardware, and safely abort the process.
- **Save / Load Preset:** Save your exact hardware configuration (MIDI channels, load waits, and naming conventions) to a **.json** file so you can instantly recall your setup the next time you plug in that specific synthesizer.

# Sampler Auto Threshold

## Introduction

The Auto Sampler Threshold Tool is an automated listening and capturing engine designed to streamline hardware sampling. Instead of manually recording long takes and slicing them by hand, this tool "listens" to your audio input, automatically starts recording when you play a note, and stops when the audio tail decays.

It then automatically processes the audio (Normalization, DC Kill) and uses advanced Music Information Retrieval (MIR) to embed loop points directly into the file metadata.

## File Settings

Define where and how your recorded files are saved.

- **Output Folder:** Click **Folder** to select the destination directory for your samples.

- **Base File Name:** Enter a prefix for your samples (e.g., "Deep Space Pad"). The tool will automatically increment the file name for each new recording (e.g., "Deep Space Pad 1.wav", "Deep Space Pad 2.wav") so you can play multiple notes without overwriting previous takes.

## Audio Settings (The Threshold Engine)

These controls determine exactly when the tool starts and stops capturing audio.

- **Channels:** Select your input source. Choose **Stereo** to preserve both channels, or **Left/Right** to capture a mono signal from a specific input.
- **Start Threshold:** The volume level (in dB) required to "wake up" the tool and begin recording. Set this slightly above your system's noise floor so that playing a note instantly triggers the capture.
- **Stop Threshold:** The volume level where the recording ends. Once the audio decays below this level and holds there for 1 full second, the capture completes and processing begins.

## Process Settings

These operations are applied non-destructively to the captured audio immediately after the stop threshold is met.

- **DC Kill:** Analyzes the waveform and removes any zero-offset bias, ensuring the waveform is perfectly centered.
- **Normalize:** Amplifies the recorded audio so its highest peak hits exactly -1.0dB, ensuring consistent volume across all samples.
- **Find Loop:** When enabled, the tool runs an MIR (Music Information Retrieval) algorithm to analyze the timbral "fingerprint" of the audio tail. It searches for two moments in the sound that match harmonically, snaps them to the nearest zero-crossings to prevent clicks, and writes these exact coordinates into the `.wav` file's `smp1` chunk.
  - **Crossfade:** Applies a 10ms crossfade between the loop end and start points.
  - **Crop Loop:** Deletes all audio outside of the detected loop, exporting only the looping section.

## Recording Settings

Configure the export format of your final `.wav` files.

- **Bit Depth:** Choose between 16-bit (int), 24-bit (int), or 32-bit (float).
- **Sample Rate:** Select the target sample rate (44100 Hz up to 96000 Hz). If your input device is running at a different rate, the tool will automatically apply high-quality resampling before saving.

## Workflow Guide

1. **Set your parameters:** Dial in your output folder, naming convention, and target sample rate.
2. **Save your Preset:** If you have found the perfect threshold settings for a specific hardware synth, click **Save Preset** so you can instantly recall this setup later.
3. **Arm the Sampler:** Click the **Record** button. The app state will change to *Listening*.
4. **Play:** Strike a chord or note on your synthesizer. The tool will instantly detect the audio passing the Start Threshold and begin capturing.
5. **Wait for the Tail:** Release the note and let the sound naturally decay. Once it drops below your Stop Threshold for 1 second, the UI will briefly pause while it processes the loop points and writes the file.
6. **Keep Playing:** The tool immediately re-arms itself (*Listening*). You can immediately play your next note to capture the next sample in the series.
7. **Stop:** When your sampling session is complete, click the red **Stop** button to disarm the tool.

## Sampler Batch Mixer

### Introduction

The Sampler Batch Mixer combines up to three audio sources with independent highpass and lowpass filtering. It features automatic mono-to-stereo upmixing, length synchronization (automatically trimming layers to match the shortest source), and Music Information Retrieval (MIR) loop extraction.

### Interface

#### Source Controls (1-3)

- **Active:** Includes the source in the mix.
- **Filters:** Enables the HP/LP filters for the source. If disabled, filters are bypassed and excluded from batch randomization.
- **Folder / File:** Select the parent directory and the specific **.wav** file to load.
- **Highpass:** 1-pole filter ranging from 20 Hz to 2 kHz to roll off low-end frequencies.
- **Lowpass:** 2-pole filter ranging from 2 kHz to 20 kHz to tame harsh highs.

#### Processing Settings

- **Output Destination:** Set the save folder and the base filename prefix. Files auto-increment to prevent overwriting.
- **Kill DC:** Removes DC offset bias from the final mixed audio.
- **Normalize:** Peak normalizes the final mixed audio to -1.0 dB.

- **Find MIR Loop:** Analyzes the final mix and embeds standard SMPL loop points directly into the `.wav` header.
  - **Crossfade:** Applies a 10ms crossfade between the loop end and start points.
  - **Crop Loop:** Deletes all audio outside of the detected loop, exporting only the looping section.
- **Bit Depth & Sample Rate:** Sets the target export format (up to 32-bit float / 96 kHz). Polyphase resampling is applied automatically if sources differ from the target rate.

## Batch & Execution

- **Generate Single:** Mixes and exports exactly the files and filter settings currently selected in the interface.
- **Execute Batch:** Recursively scans the selected source folders to generate randomized audio combinations.
  - **Batch Count:** The number of files to generate per execution.
  - **Randomize Filters per Pass:** Randomly sweeps the HP and LP filters for every generated file.
  - *Note on Batch Logic:* The engine uses a "Weighted Anchor" system. It selects a random file from Source 1, then automatically evaluates Sources 2 and 3 to find files with the closest matching durations to ensure cohesive layers.

## Presets

- **Save / Load:** Stores and recalls all UI parameters to a file, including active sources, exact filter values, selected target folders, and global processing settings.

# Sampler Batch Resonator

## Introduction

This tool bridges the gap between basic audio mixing and advanced procedural sound design. By utilizing Fast Fourier Transform (FFT) Convolution, the Resonator allows you to mathematically imprint the acoustic and tonal characteristics of one audio folder onto another, generating entirely new hybrid textures, rhythmic top-loops, and complex cinematic drones perfectly tailored for your next Kontakt instrument or synth expansion.

## The Core Concept

Convolution relies on multiplying two audio signals together in the frequency domain. To get the most out of this tool, it helps to think of your two sources in specific roles:

- **Source A (The Exciter):** Think of this as the "driver." This is typically your rhythmic, transient, or dominant audio source (e.g., a drum break, a vocal chop, or a synth pluck).

- **Source B (The Resonator):** Think of this as the "environment" or "body." This audio provides the tonal, spatial, or textural characteristics that will be forced onto the Exciter (e.g., a metallic crash, a lush reverb tail, or a dissonant drone).

## Source Configuration & Pre-Filtering

Before the convolution math happens, you have precise control over the input signals to prevent frequency buildup and muddiness.

- **Folder Drag & Drop:** Simply drag and drop any folder directly into the Source A or Source B text boxes to load your audio directories.
- **File Selection:** Use the dropdown menu to manually select a specific file for single-generation testing, or leave it as-is to let the Batch engine pull from the entire folder.
- **Highpass (HP) & Lowpass (LP) Filters:** \* *Pro-Tip:* Convolution multiplies amplitudes. If both your Exciter and Resonator have heavy low-end frequencies, the resulting audio will be pure mud. Use the Highpass slider (especially on Source B) to roll off frequencies below 200Hz to keep your mix clean and punchy.

## Convolution Parameters

This section dictates how the Resonator shapes the impulse before blending it with your original sound.

- **Reverse Resonator Impulse:** Flips the audio array of Source B *before* convolution. This is incredible for creating forward-swelling textures, reverse-reverbs, and haunting build-ups.
- **Crop Mix to Exciter Length:** Because convolution mathematically adds the length of both files together, convoluting a 1-second drum loop with a 10-second drone normally creates an 11-second file. Checking this box forcefully truncates the final output to perfectly match the original length of Source A, saving massive amounts of CPU and RAM while keeping your loops tight.
- **Dry/Wet:** Blends your original, unaffected Exciter (Dry) with the newly convolved hybrid signal (Wet). This utilizes an Equal-Power trigonometric crossfade, ensuring your volume never unexpectedly dips at the 50% mark.
- **Res Decay:** Applies a strict volume fade-out to the Resonator *before* convolution occurs. If you want to use a massive 2-minute atmospheric drone as your Resonator but only want a tight, 500ms metallic tail on your drum hits, enable this and set the slider to 500ms.

## Processing & MIR Loop Extraction

Once the convolution and blending are complete, the audio passes through an automated mastering and metadata pipeline.

- **Kill DC:** Automatically calculates and removes any DC offset introduced by the convolution math, centering your waveform.
- **Normalize:** Applies a global peak normalization (safely hard-clipped to 0dBFS) to maximize the volume of the hybrid sound while preserving its stereo image.
- **Find Loop:** Engages the internal Music Information Retrieval (MIR) engine. This algorithm analyzes the audio's frequency similarities to automatically locate the mathematically perfect, seamless loop points within the generated file.
  - **Crossfade:** Applies a micro-crossfade at the detected loop boundaries to guarantee a click-free transition.
  - **Crop Loop:** If enabled, the engine deletes all audio outside the detected loop points. Instead of a long file with embedded metadata, you get a perfectly truncated, ready-to-deploy looping asset.
- **Bit Depth & Sample Rate:** Select your target export formats. The engine handles all resampling automatically.

## Batch Generation

The true power of the Resonator lies in its ability to procedurally generate hundreds of unique assets in seconds.

- **Batch Count:** Select how many hybrid files you want to generate.
- **Randomize Filters:** When checked, the Batch engine will automatically generate a random, musically scaled Highpass and Lowpass filter sweep for *every single file* it generates, ensuring massive tonal variety across your exported batch.
- **Generate Single vs. Execute Batch:** Use "Generate Single" to process exactly the two files currently selected in your dropdowns. Use "Execute Batch" to let the engine randomly pair Exciters and Resonators from your selected folders.

*Note: You can instantly abort a massive batch run at any time by clicking the **Cancel** button. The engine will safely finish its current mathematical operation and unlock the UI.*

## Toolbox Music Handyman

### Introduction

The Music Handyman Toolbox is a lightweight, essential utility designed for sound designers, music producers, and DSP developers. Whether you are calculating the exact sample length of a single-cycle wavetable, tuning a synthesizer oscillator to a specific concert pitch, or mapping out a diatonic chord progression, this tool provides instant, mathematically precise conversions and music theory references without needing to open a web browser.

### Interface

To maximize screen real estate, the toolbox utilizes a custom collapsible accordion interface. It is divided into three distinct modules, allowing you to instantly expand or hide specific sets of calculations to fit your workflow. Additionally, you can save and load your current calculator states using the **Save Preset** and **Load Preset** buttons at the bottom of the tool.

## Music, Pitch, and Cycle Calculators

This section translates musical timing into technical values used in audio engineering, and features a three-column dashboard for advanced time and pitch manipulation.

- **BPM (Beats Per Minute):** Use the slider to set your project tempo, ranging from 10 to 300 BPM.
- **Div (Tempo Division):** Select the rhythmic value you want to calculate. The dropdown includes standard beats and bars, alongside straight, dotted (d), and triplet (t) subdivisions.
- **SR (Sample Rate):** Select the target sample rate of your project or audio stream (ranging from 8,000 Hz up to 192,000 Hz).

### Primary Outputs:

- **ms (Milliseconds):** The exact duration of the selected division. Perfect for setting reverb pre-delays, compressor release times, or delay feedback loops.
- **Hz (Hertz):** The frequency of the selected division. Crucial for syncing LFO rates or rhythmic modulators to the project tempo.
- **Samples:** The exact number of digital audio samples the division occupies. Essential for DSP developers setting buffer sizes or sample-accurate loop points.

### Advanced Dashboard:

- **Delay Matrix:** A rapid-reference grid that instantly displays the exact millisecond values for Straight, Dotted, and Triplet timings across 1/4, 1/8, 1/16, and 1/32 notes based on your current BPM. Ideal for programming complex, syncopated delay networks.
- **Repitch (Pitch Shifting):** Input an original BPM and a new target BPM. The output tells you exactly how many semitones (st) and cents (c) you need to pitch-shift an audio file so it fits the new tempo without using time-stretching algorithms.
- **Cycle Length:** Input a target frequency in Hz. The output provides the exact length of a single cycle of that frequency in both milliseconds and samples. This is vital for designing single-cycle waveforms and wavetables.

## Pitch, Tuning, and Harmonics

A high-precision pitch and frequency workstation that maps musical notes to raw audio data, and vice-versa.

- **Note:** Select any standard musical note from C0 up to B8.

- **Tuning (Hz):** Select your A4 reference pitch. While 440 Hz is the modern standard, you can instantly recalculate the entire spectrum to alternative tunings (such as 415 Hz, 432 Hz, or 442 Hz).
- **Hertz Output:** Displays the exact frequency of the chosen note down to four decimal places.
- **MIDI Note Output:** Displays the exact, standard MIDI Note Number (0–127) for the selected pitch.
- **Harmonics:** Calculates and displays the first eight overtones (H1 through H8) of your selected root note. This is a goldmine for advanced sound design, specifically for setting up FM operator ratios or programming additive synthesis parameters.
- **Rev Calc (Reverse Pitch Calculator):** Input a raw frequency (Hz), and the tool calculates the closest musical note and the exact tuning offset in cents. Perfect for tuning unruly analog drum samples or analyzing resonant frequencies in field recordings.

## Musical Scales, Chords, and Progressions

A rapid music theory reference and composition assistant for mapping out scales, modes, and harmonies.

- **Root:** Select the fundamental root note of the scale (C through B).
- **Scale:** Choose from 31 different musical scales and modes. This includes Western standards (Major, Minor, Harmonic Minor), Greek modes (Dorian, Phrygian, Lydian), Jazz scales (Bebop Dominant, Blues), and global/exotic scales (Hirajoshi, Hungarian Minor, Persian, Balinese).
- **Add 7ths (Quadrads):** A toggle switch that dictates how the chord generator builds its harmonies. When checked, the tool expands standard three-note triads into complex four-note 7th chords.

## Outputs:

- **Notes:** Instantly displays the exact notes contained within the selected scale in a clear, comma-separated format. Perfect for referencing while programming arpeggiators or locking MIDI effects to a specific key.
- **Chords:** Generates and displays the naturally occurring diatonic chords for every degree of the chosen scale.
- **Progressions:** Automatically populates a series of standard, widely used chord progressions (such as 1-4-5 and 2-5-1) using the exact diatonic chords generated from your selected root and scale. Instantly provides a harmonic roadmap when writing pads or breaking out of a creative rut.

## Toolbox RIFF Reader

## Introduction

The Toolbox RIFF Reader is a lightning-fast diagnostic inspector for `.wav` files. Rather than loading heavy audio data into memory, the engine scans and parses only the RIFF chunk headers and metadata payloads. This allows for instantaneous inspection of loop points, broadcast extensions, and deeply embedded Native Instruments NKS configurations, even on massive audio files.

## File Navigation

- **Folder Selection:** Click "Select Folder" to target a directory. The tool will recursively scan for and catalog all available `.wav` files if recursive is checked, and otherwise scan the base folder.
- **Recursive:** Toggles between scanning the base folder alone, or the base folder and any found sub folders.
- **Target File:** A dropdown listing all valid audio files in the targeted folder. Selecting a file instantly parses its internal RIFF structure.

## Display Modes

The parsing engine can translate the binary chunk data into three different viewing formats using the **Display Mode** dropdown:

- **Readable (Deep Parse):** Translates known binary chunks into human-readable parameters.
  - Parses standard `fmt` properties (sample rate, bit depth, channels).
  - Decodes `smpl` chunks to verify loop start/end points and modes.
  - Decodes `inst` chunks for root note and fine-tuning data.
  - Unpacks `LIST` and `bext` (Broadcast Extension) strings.
  - **ID3 & NKS Extraction:** Aggressively parses ID3v2 tags (handling sync-safe byte alignments) to extract text frames, comments, and deeply embedded `GEOB` / `PRIV` payloads, specifically hunting for and extracting Native Instruments NKS JSON arrays.
  - *Fallback:* For unknown chunks, it automatically extracts and lists printable ASCII strings.
- **HEX Raw:** Bypasses translation and displays the raw hexadecimal byte stream for every non-audio chunk (capped at 4KB for UI safety).
- **ASCII Raw:** Bypasses translation and displays the raw ASCII character stream for every non-audio chunk, substituting non-printable bytes with dots.

## The Display Tree

The main interface uses a hierarchical tree structure. Click the arrows next to a chunk (e.g., `<smpl>` or `<ID3>`) to expand it and view the extracted properties, frames, or raw data payloads inside. The heavy `<data>` chunk is explicitly bypassed and summarized by its byte size to ensure the application never stalls.

## Presets

- **Save / Load:** Captures your current workspace state - including the targeted folder path, the specifically selected file, and your active Display Mode to a preset file for quick recall.

## Toolbox Sampleset Check

### Introduction

The Toolbox Sampleset Check is an advanced Quality Assurance utility. It recursively scans directories of `.wav` files and flags any assets that deviate from the expected standard. It is designed to catch rendering glitches, misplaced files, and formatting errors before a sample pack or instrument goes into production.

### How to Use

1. **Browse Folder:** Select the root directory of your project. The tool will scan the root folder and all subfolders recursively.
2. **Set Metrics & Tolerances:** Enable only the specific checks you need for your current project phase.
3. **Save/Load Presets:** Use the preset buttons at the bottom right to save your current tolerance profile (e.g., "Strict Drum Kit", "Loose Ambience") for future use.
4. **Run QA Scan:** Click the main scan button on the bottom left. The tool will execute either a Fast Scan or Deep Scan depending on your selected parameters, yield progress on the loading bar, and post all identified errors directly to the main application's console log.

### Smart Scanning: Fast vs. Deep Scan

To save processing time, the tool intelligently analyzes your selected metrics before running:

- **Fast Scan (Headers Only):** If you are only checking structural integrity (Sample Rate, Bit Depth, Length, Channels, Loops, or Duplicates), the tool reads the binary file headers without loading audio data into RAM. This can scan thousands of files in seconds.
- **Deep Scan (DSP Enabled):** If you enable RMS or Peak Deviance checks, the tool will load the audio arrays into memory to perform the necessary digital signal processing math.

## The Checks & Rulesets

### Deviance Checks (The Two-Pass Method)

For audio volume, hardcoding targets is rarely effective due to the dynamic nature of sound. Instead, the tool uses a "Two-Pass" deviance method:

- **RMS Deviance:** In Pass 1, the tool calculates the mathematical median RMS of the entire folder. In Pass 2, any file that strays further from the median than your allowed **Max dB Diff** is flagged.
- **Peak Deviance:** Follows the same logic, identifying outlier transients and potential volume inconsistencies across the set.

### Structural Integrity

- **Minimum Length:** Flags any file shorter than the set second duration. This is highly effective at catching 0-byte ghost files or aborted DAW exports.
- **Sample Rate & Bit Depth:** Enforces strict adherence to standard encoding formats (e.g., 44100Hz, 24-bit).
- **Channel Count:** You can enforce strict Mono or Stereo compliance, or select "Match Median" to allow the tool to enforce whichever format is dominant in the folder.

### Logic Checks

- **Loop Points:** Verifies the presence (or strict absence) of embedded **smp1** loop points across the audio files.
- **Duplicate Filenames:** Recursively checks all subdirectories. If two files share the exact same name anywhere in the file tree, an error is logged specifying both locations, preventing overwrite conflicts in modern samplers.

## Toolbox Smart Renamer

### Introduction

The Toolbox Smart Renamer is a precision batch-renaming utility designed to handle massive libraries of samples, presets, and project folders. It features a highly predictable, linear

processing engine with a live, color-coded preview system, ensuring that you always know exactly what will happen to your files before you commit.

Built with data integrity as its primary focus, the tool operates in a completely non-destructive mode by default, mathematically reconstructing your original folder hierarchies into isolated output directories. For advanced users, a heavily safeguarded destructive overwrite mode is available, protected by intelligent sibling-chain detection and directory-overwrite locks to guarantee zero data loss.

## Interface & Controls

### Targeting & Filtering

Define exactly which files or folders the engine should look at before applying any naming logic.

- **Select Folder:** Opens an OS dialog to choose the root directory containing the assets you want to rename. You can also drag and drop a folder directly anywhere onto the tool interface.
- **Recents:** A quick-access dropdown menu that automatically remembers the last 10 directories you worked in.
- **Recursive:** When checked, the tool will dive into all subdirectories within your selected root folder. The engine intelligently ignores previous **\_Renamed** output folders to prevent infinite loops.
- **Target:** Choose whether the renaming logic should apply strictly to **Files**, **Folders**, or **Files & Folders**.
- **Extension Filter:** A comma-separated filter (e.g., **wav**, **aif**, **json**) to target specific file types. Leave this blank to target all files.

### Operation Logic

Select the mathematical string operation you wish to apply to your targets. The interface will dynamically update to show only the relevant parameters for your chosen operation.

- **String Replace:** Search for specific text and replace it. Controlled by the **Find** and **Replace** text boxes.
- **Add Prefix / Add Suffix:** Attaches the exact text from the **Text** input box to the beginning or end of the filename.
- **Remove special chars:** Strips all characters except standard alphanumeric text, dashes, underscores, spaces, and periods.
- **Remove digits:** Strips all numbers (0-9) from the filename.
- **To lower case / To upper case / To Title Case:** Standard casing conversions.

- **Break CamelType:** Intelligently splits CamelCase words (e.g., **DeepBass** becomes **Deep Bass**). You can choose to break the words using a Space, Underscore, or Dash, and optionally force the resulting words to lowercase.
- **Add running prefix / Add running suffix:** Generates a sequential numerical count based on the natural alphabetical order of your folder. It exposes a **Start** parameter to define the beginning number, and an **Append With** dropdown to seamlessly place a space, underscore, or dash between the sequence and your filename.

## Positional Conditions

For operations like String Replace and character/digit removal, you can define exactly where the logic applies using the **Where** dropdown:

- **Anywhere:** Applies the operation globally across the entire string.
- **At start / At end:** Restricts the operation to the absolute beginning or end of the filename.
- **Before / After underscore, space, or dash:** Restricts the operation to text occurring strictly before or after the first instance of the specified delimiter.

## Casing Targets

When applying casing operations to files, use the **Apply To** dropdown to prevent accidentally breaking file extensions:

- **File Name:** Changes the casing of the core name, leaving the **.extension** untouched.
- **File Extension:** Changes the casing of the extension only (e.g., turning **.WAV** into **.wav**).
- **Extension and Name:** Applies the casing rules to the entire string.

## The Preview Engine

The core of the Smart Renamer is its live, multi-threaded WYSIWYG (What You See Is What You Get) table. It populates instantly as you type or adjust parameters, parsing up to 100,000 items.

## Color Coding & Collision Detection

The table provides immediate visual feedback on the state of every file:

- **White/Gray Text:** The file matches your filters, but your current operation does not change its name. (These files are ignored during execution).
- **Green Text:** The file name will be successfully changed.

- **Orange Text (Soft Collision):** Occurs when Overwrite mode is enabled and the destination file already exists. The tool will allow execution, but warns you that data will be replaced.
- **Red Text (Hard Collision):** A critical safety block. Occurs if two files are trying to rename to the exact same destination, if you are attempting to overwrite a directory, or if a "domino effect" is detected (where a file attempts to overwrite a sibling file that is also queued for renaming). **Execution is disabled until Hard Collisions are resolved.**

## Execution & Output

Once your preview is perfectly dialed in, click **Execute Rename** to process the batch.

- **Non-Destructive Mode (Default):** When the Overwrite checkbox is left unchecked, the tool will never touch your original files. It creates a brand new **\_Renamed** folder next to your source data. It dynamically mimics your exact internal folder structures and copies the freshly renamed files into it. If you run multiple passes, the tool automatically versions the output folders (e.g., **\_Renamed\_1**, **\_Renamed\_2**) to prevent silently erasing your previous work.
- **Overwrite Files (Destructive):** When checked, the tool renames the files and folders directly in place on your hard drive.
- **Bottom-Up Processing:** The engine natively processes the deepest nested files first before renaming parent folders. This structural guarantee prevents "path not found" crashes during complex, recursive folder renames.

## Presets

- **Save Preset / Load Preset:** Save your exact operation logic, targeting rules, filters, and overwrite states to a file.

# Wavetable Converter

## Introduction

The Ocean Swift Wavetable Converter allows you to effortlessly convert wavetables between different formats and frame sizes. This ensures your custom sounds and favorite wavetable packs are perfectly compatible across all your synthesizers and modules. All output files maintain their original amplitude fidelity and are scaled to peak levels for consistent volume.

## Process

The converter utilizes a streamlined, folder-centric workflow. You can load a directory using the "Select Folder" button, choose from your "Recent" history, or simply drag and drop a folder directly onto the preview area. If you drop a single audio file, the tool will smartly load its parent folder and automatically select that specific file for you. Once loaded, all valid audio files (.wav, .aif, .aiff) are scanned and populated into the Target File dropdown so you can easily navigate and preview their waveforms.

When you are ready to convert, you can choose to process only the currently viewed file using "Export Single", or batch process the entire loaded directory using "Export Folder". For single exports, a save dialog determines the resulting file name. For folder exports, a dialog determines the base output destination. When "All" is selected for the converted frame size during a folder export, subfolders are automatically created for each possible frame size. In each of these subfolders, a FolderInfo.txt file is created which contains the frame size for easier handling in Serum. The resulting file names are derived from the original file name and the conversion details.

A loop point after the first cycle is embedded in each created file, helping in marking the frame size within the file itself for easier handling in synths such as Surge, VAZ Modular, and others.

If the frame size for both the loaded file and export is the same (or when the frame sizes match during an "All" operation) the wavetable is created with a loop point at the frame size, but not interpolated.

The resulting files are exported as .wav files with the selected sample rate and bit depth. Your previous workspace (last loaded folder and file) is automatically remembered and restored the next time you launch the tool.

## Controls

- **Select Folder:** Choose a folder to scan for .wav, .aif, and .aiff files to be processed.
- **Recent:** Open a dropdown menu to quickly reload up to 10 recently used folders.
- **Target File:** Select a specific file from the loaded folder to preview in the visualizer or to process using Export Single. Use the < and > buttons to quickly navigate sequentially through the files.
- **Original Frame Size:** Set the frame size of the original wavetable file(s). The correct original frame size must be known and set in order for the conversion to work correctly.
- **Convert Frame Size:** Select the target frame size for exporting. Selecting "All" will export files for all available frame sizes.
- **Sample Rate:** Select the sample rate for the exported files.
- **Bit Depth:** Select the bit depth for the exported files.
- **Export Single:** Initiate the interpolation process and create a new wavetable file exclusively for the currently selected Target File.

- **Export Folder:** Batch process and export all valid wavetable files found within the currently loaded folder.
- **Cancel:** Cancels an active interpolation and file creation process.
- **Clear:** Clears the currently loaded folder and resets the visualizer.
- **Save Preset / Load Preset:** Save and recall your specific conversion configurations (frame sizes, sample rate, and bit depth) to JSON files for quick setup in the future.

# Wavetable Creator

## Introduction

The Ocean Swift Wavetable Creator is a versatile and intuitive software designed for musicians, producers, and sound designers seeking the ultimate level of creativity in their sonic explorations. The application provides a powerful environment to generate custom wavetables, blend waveforms, and create evolving, rich sounds that can be used in wavetable synthesizers. The robust export options ensure that the resulting wavetables are compatible with almost any possible wavetable synthesizer.

Ocean Swift Wavetable Creator allows you to morph and blend up to two different waveforms into new, unique timbres that can be further customized with advanced processing options. The application features a user-friendly graphical interface where users can select from a variety of waveforms, including classic shapes like sine, square, triangle, and sawtooth, as well as more complex and unusual waveforms such as additive, FM, and even custom noise-based waveforms.

A core strength of the Wavetable Creator is its interaction system, which provides multiple ways to combine two sets of waveforms creatively. Users can apply crossfades, mixes, amplitude modulation, and harmonic blending options to create wavetables that evolve and transform over time. For sound designers interested in more experimental techniques, there are unique blending options like ring modulation, morphing phase shifts, and harmonic emphasis, providing a wealth of possibilities to craft evolving sounds with intricate harmonic content.

Ocean Swift Wavetable Creator also supports advanced waveform processing tools, such as clipping, tube saturation, folding, quantizing and more. Each processing effect can be applied individually to either waveform, allowing for complex sound manipulations that produce truly unique sonic results.

In addition, the app offers a batch creation feature that makes it possible to automate the generation of large sets of wavetables with varied settings. Users can randomize waveform choices, processor effects, and blending methods to generate a library of unique wavetables that can be used in different projects.

Whether you are an experienced sound designer or a producer exploring new possibilities for custom synthesizer sounds, Ocean Swift Wavetable Creator aims to empower your creativity. With the ability to produce wavetables ranging from subtly morphing textures to aggressive, experimental soundscapes, this tool is an essential addition to any producer's toolkit. The straightforward interface makes it accessible to beginners, while the depth of modulation and processing capabilities ensures that professionals will find a wealth of creative options to explore.

## Waves

At the core of the Wavetable Creator are two separate waves that are each constructed by morphing between five selectable wave shapes through the length of the wavetable. After each wave is created, it can be further shaped by waveshaping processors, and these two waves can interact in a variety of ways in order to construct the final wavetable.

## Wave Shapes

The Wavetable Creator offers a wide selection of wave shapes that serve as building blocks for your custom wavetables. Clicking to select a shape in each of the wave shape menus while holding SHIFT will set all the shapes for the respective wave to that selected shape.

### Sine

**Sine:** A pure, smooth, and continuous wave that represents a single frequency. Ideal for clean and simple tones.

**Exponential Sin:** A sine wave that varies in frequency exponentially over time, creating an evolving pitch sweep.

**Sine FM 1:** A basic sine wave subtly modulated by a secondary frequency. This introduces a slight frequency modulation effect, giving the sound a touch of movement and tonal complexity. Ideal for adding a bit of character without overwhelming the fundamental tone.

**Sine FM 2:** A sine wave with moderate frequency modulation. This setting enhances the sine wave's harmonic content, resulting in a richer and more dynamic sound, great for subtle textures in ambient or atmospheric sound design.

**Sine FM 3:** A sine wave with strong frequency modulation, creating pronounced harmonic complexity.

**Sine Self FM 1:** A sine wave that slightly modulates itself.

**Sine Self FM 2:** A sine wave that moderately modulates itself.

**Sine Self FM 3:** A sine wave that strongly modulates itself.

**Sine Tri:** Morphs gradually from a sine wave to a triangle wave, creating a transitional, evolving waveform.

**Sine Square:** A sine wave with its amplitude squared, emphasizing smoother peaks and reducing sharpness.

**Double Sine:** A sine waveform completing two cycles within the frame size, resulting in a sound that is one octave higher than a single-cycle sine wave. Useful for creating higher-pitched sine tones with the same smooth characteristics.

## Tri

**Tri:** A linear waveform with a softer sound than a square wave. It produces a tone rich in odd harmonics but softer in timbre.

**Tri BL:** Band-limited version of the Tri waveform, useful for reducing aliasing.

**Tri Saw:** A combination of triangle and sawtooth waves for a blend of sharp and smooth characteristics.

**Tri Saw BL:** Band-limited version of the Tri Saw waveform, useful for reducing aliasing.

**Double Tri:** A triangle waveform with two cycles per frame, creating a higher-pitched sound while retaining the characteristic smoothness of the triangle wave.

**Double Tri BL:** Band-limited version of the Double Tri waveform, useful for reducing aliasing.

## Saw

**Saw:** A sharp, ramping waveform that has both even and odd harmonics, giving it a bright, rich sound. Great for leads and basses.

**Saw BL:** Band-limited version of the Saw waveform, useful for reducing aliasing.

**Exp Saw:** A variation of the sawtooth wave with an exponentially decaying ramp, adding a unique timbre.

**Exp Saw BL:** Band-limited version of the Exp Saw waveform, useful for reducing aliasing.

**Shark Fin:** A wave that ramps up quickly and then drops, resembling the profile of a shark's fin. Used for unique, edgy tones.

**Shark Fin BL:** Band-limited version of the Shark Fin waveform, useful for reducing aliasing.

**Double Saw:** A sawtooth waveform that completes two cycles within the frame size, producing an octave-higher tone while maintaining the aggressive harmonic profile of the sawtooth.

**Double Saw BL:** Band-limited version of the Double Saw waveform, useful for reducing aliasing.

## Square

**Square:** Alternates abruptly between high and low values, creating a sharp, buzzy sound. Useful for electronic and chiptune sounds.

**Square BL:** Band-limited version of the square waveform, useful for reducing aliasing.

**Square Tri:** A blend of square and triangle waveforms, balancing the sharpness of the square with the smoothness of the triangle.

**Square Tri BL:** Band-limited version of the Square Tri waveform, useful for reducing aliasing.

**Square Saw:** A combination waveform with aspects of both square and sawtooth waves for a unique harmonic profile.

**Square Saw BL:** Band-limited version of the Square Saw waveform, useful for reducing aliasing.

**Double Square:** A square waveform that completes two cycles within the frame, doubling its frequency to produce a sound an octave higher with a strong, distinct pulse.

**Double Square BL:** Band-limited version of the Double Square waveform, useful for reducing aliasing.

## Pulse

**Narrow Pulse 1:** A pulse wave with a short duty cycle, creating a thin, sharp tone.

**Narrow Pulse 1 BL:** Band-limited version of the Narrow Pulse 1 waveform, useful for reducing aliasing.

**Narrow Pulse 2:** A slightly wider pulse than Narrow Pulse 1, producing a distinct sharpness with more harmonic content.

**Narrow Pulse 2 BL:** Band-limited version of the Narrow Pulse 2 waveform, useful for reducing aliasing.

**Fat Pulse 1:** A pulse wave with a broader duty cycle, offering a thicker and warmer tone.

**Fat Pulse 1 BL:** Band-limited version of the Fat Pulse 1 waveform, useful for reducing aliasing.

**Fat Pulse 2:** An even wider pulse wave that creates a fuller and more robust sound.

**Fat Pulse 2 BL:** Band-limited version of the Fat Pulse 2 waveform, useful for reducing aliasing.

**Double Pulse:** A pulse wave with two peaks per cycle, adding complexity to the sound.

**Double Pulse BL:** Band-limited version of the Double Pulse waveform, useful for reducing aliasing.

## Noise

**Noise:** Random signal with equal energy at all frequencies, producing a "static" or "hiss" sound. Useful for effects and percussive elements.

**Noise BL:** Band-limited (filtered) version of the Noise waveform, useful for reducing aliasing.

**Pink:** A noise signal with energy decreasing proportionally with frequency, producing a warm, natural sound often used in soundscapes.

**Pink BL:** Band-limited version of the Pink waveform, useful for reducing aliasing.

**Brown:** A deeper noise with an even stronger emphasis on low frequencies, simulating natural phenomena like waterfalls or thunder.

**Brown BL:** Band-limited (filtered) version of the Brown waveform, useful for reducing aliasing.

## Harmonics

**Harmonic Series 3:** A composite waveform including the first three harmonics, creating a richer sound.

**Harmonic Series 4:** Adds the first four harmonics, increasing the complexity and richness of the sound.

**Partials 1 3:** A mix of the fundamental and the third harmonic, producing a distinctive tone.

**Partials 1 4:** Combines the fundamental and the fourth harmonic for a unique tonal blend.

**Partials 1 5:** Incorporates the fundamental and fifth harmonic, adding subtle brightness.

**Partials 1 7:** Incorporates the fundamental and fifth harmonic, adding sparkling brightness.

**Partials 1 3 5:** A combination of the fundamental, third, and fifth harmonics for a more complex sound.

**Partials 1 3 8:** A combination of the fundamental, third, and eighth harmonics, producing a sound with both warmth and a slightly metallic edge due to the higher partial.

**Partials 1 4 7:** A blend of the fundamental, fourth, and seventh harmonics, creating a harmonic-rich waveform.

**Partials 1 6 9:** A blend of the fundamental, sixth, and ninth harmonics, creating a harmonic-rich waveform.

**Partial 3->1 FM:** A sine wave whose fundamental is frequency modulated by the 3rd partial.  
**Partial 4->1 FM:** A sine wave whose fundamental is frequency modulated by the 4th partial.  
**Partial 5->1 FM:** A sine wave whose fundamental is frequency modulated by the 5th partial.  
**Partial 6->1 FM:** A sine wave whose fundamental is frequency modulated by the 6th partial.  
**Partial 7->1 FM:** A sine wave whose fundamental is frequency modulated by the 7th partial.  
**Partial 8->1 FM:** A sine wave whose fundamental is frequency modulated by the 8th partial.  
**Partial 3->1 AM:** A sine wave whose fundamental is amplitude modulated by the 3rd partial.  
**Partial 4->1 AM:** A sine wave whose fundamental is amplitude modulated by the 4th partial.  
**Partial 5->1 AM:** A sine wave whose fundamental is amplitude modulated by the 5th partial.  
**Partial 6->1 AM:** A sine wave whose fundamental is amplitude modulated by the 6th partial.  
**Partial 7->1 AM:** A sine wave whose fundamental is amplitude modulated by the 7th partial.  
**Partial 8->1 AM:** A sine wave whose fundamental is amplitude modulated by the 8th partial.

## Randoms

**Additive Random:** This waveform starts with a fundamental sine wave and randomly adds between 2 to 8 harmonics, selected from the first 16 harmonics. Each harmonic is mixed with a randomized volume, which gradually decreases with each additional harmonic. This creates a unique and evolving harmonic structure, adding subtle complexity to the sound.

**Additive Random FM:** This waveform is similar to Additive Random but introduces subtle frequency modulation to each added harmonic. As each harmonic is layered, a slight pitch modulation creates evolving textures and adds richness, making it ideal for dynamic, atmospheric sounds with a hint of unpredictability.

**Complex Random:** This waveform is constructed by randomly selecting and combining three waveforms (Sine, Triangle, Sawtooth, Narrow Pulse, or Square). Each chosen waveform is given a random volume, producing a complex, layered shape with a distinct timbre that varies with each selection.

**Complex Rand BL:** Band-limited version of the Complex Random waveform, useful for reducing aliasing.

**Phased Random:** Similar to the Complex Random waveform, but with an added twist—each selected waveform is given a random phase shift. This adds an extra layer of modulation, creating subtle variations in the waveform's structure and introducing dynamic phase interactions between the combined shapes.

**Phased Rand BL:** Band-limited version of the Phased Random waveform, useful for reducing aliasing.

**Complex Doubles:** This waveform functions similarly to the Complex Random waveform, but with an expanded selection, including both standard and "Double" versions of each wave shape. It randomly selects and combines three waveforms from the following options: Sine, Triangle, Sawtooth, Square, Narrow Pulse, Double Sine, Double Triangle, Double Sawtooth, and Double Square. Each chosen waveform is assigned a random volume, creating a rich, layered timbre that varies with each selection. The inclusion of "Double" waveforms adds higher harmonics by completing two cycles within the same frame, resulting in a more complex and harmonically rich sound ideal for creating intricate textures and evolving soundscapes.

**Comp Doubles BL:** Band-limited version of the Complex Doubles waveform, useful for reducing aliasing.

## Special

**TX2:** This unique waveform, inspired by Yamaha's TX2 shape, features a quick ramp-up to a sharp peak, followed by a steep drop to a pronounced dip before returning to zero.

**TX2 Gaussian:** A unique hybrid waveform that combines the characteristics of two distinct shapes. The first half of the waveform follows the profile of the TX2 shape, featuring a sharp triangular peak and descent, emulating the classic TX2 structure. This is followed by a smooth Gaussian Bell curve in the second half, introducing a rounded peak and gradual tapering.

**Gaussian Wave:** A wave shaped like a Gaussian curve, ideal for softer, bell-like sounds and smooth transitions.

**Chirp:** A waveform with linearly increasing frequency over time, producing a "sweeping" sound effect.

**Exp Chirp:** A waveform with exponentially varying frequency, creating a more rapid sweep effect.

**Impulse (Sinc):** A wave that resembles an impulse response, ideal for sharp, percussive sounds and unique sound design.

**Double Impulse:** Two evenly spaced impulses.

**Tangent:** A waveform based on the hyperbolic tangent function, providing a soft clipping effect. This shape introduces subtle harmonics, resulting in a warmer, slightly compressed tone.

**Damped:** A decaying sine wave oscillator that simulates a damped oscillation. This waveform gradually loses amplitude, creating a ringing effect, ideal for simulating natural, percussive, or plucked instrument sounds.

## Wave Shapers

For either Wave 1 or Wave 2, two shaper processor stages can be selected which further modulate, shape and manipulate the wave. A final processor stage can be applied to the final wave resulting from the interaction step.

### Off

**None:** Applies no processing to the waveform, maintaining its original state.

## Filters

**LP Nyquist:** Applies a low-pass filter with a cutoff frequency set near the Nyquist frequency, effectively removing high-frequency content above this threshold. This process ensures a cleaner waveform and reduces aliasing artifacts.

**LP 1/2 Nyquist:** Applies a more aggressive low-pass filter with a cutoff frequency set to half of the Nyquist frequency. This filter removes a broader range of high frequencies, resulting in a smoother waveform with reduced harmonic content.

**LP 1/4 Nyquist:** Applies a more aggressive low-pass filter with a cutoff frequency set to one fourth of the Nyquist frequency. This filter removes an even broader range of high frequencies, resulting in a smoother, rounded waveform with vastly reduced harmonic content.

## Shift

**Invert:** Inverts the polarity of the waveform.

**Phase 30deg:** Shifts the waveform phase by 30 degrees.

**Phase 45deg:** Shifts the waveform phase by 45 degrees.

**Phase 90deg:** Shifts the waveform phase by 90 degrees.

## Shapers

**Log:** Applies a logarithmic transformation to the waveform, compressing its dynamic range and making softer parts more prominent while maintaining overall character.

**Exp:** Applies an exponential curve to the waveform, emphasizing louder sections and creating a sharper, more aggressive sound profile.

**Bend Up:** Applies an upward bending to the waveform, emphasizing higher values and creating a more pronounced peak in the positive direction.

**Bend Down:** Applies a downward bending to the waveform, enhancing lower values and creating a pronounced peak in the negative direction.

**Shape Soft:** Applies a subtle shaping to the waveform, enhancing the contour while maintaining the overall shape. It smooths and gently emphasizes certain features.

**Shape Medium:** More pronounced shaping that adds noticeable contour and emphasis to the waveform's structure, resulting in a richer sound.

**Shape Hard:** Aggressive waveform shaping that alters the original signal significantly, creating a bold and uniquely textured sound.

**Fold:** Emphasizes the waveform's peaks by folding them back on themselves, creating a more complex harmonic structure as the amplitude exceeds certain limits.

## Distortions

**Clip Soft:** Softly limits the waveform's amplitude, producing gentle distortion. This processor preserves more of the waveform's dynamics while limiting extreme peaks.

**Clip Medium:** Moderately limits the waveform's amplitude for a more pronounced clipping effect, adding more distortion than the soft variant.

**Clip Hard:** Applies aggressive clipping to the waveform, sharply limiting the amplitude and introducing significant distortion for an intense, gritty sound.

**Tube Soft:** Simulates the warmth of a soft tube saturation, adding subtle harmonic enhancement that enriches the waveform's sound without heavy distortion.

**Tube Medium:** Introduces more pronounced tube saturation, producing a warmer, fuller sound with noticeable harmonic overtones.

**Tube Hard:** Strong tube saturation for a heavily distorted and rich sound, adding significant harmonic content and warmth to the waveform.

## Decimators

**Quantize Soft:** Gently reduces the resolution of the waveform, adding subtle digital artifacts that create a lo-fi effect without overwhelming the original sound.

**Quantize Medium:** Applies a moderate level of quantization, introducing more digital artifacts and a noticeable lo-fi texture.

**Quantize Hard:** Strong quantization that greatly reduces the resolution of the waveform, producing a distinct digital, lo-fi sound.

**Quantize Harsh:** The most aggressive form of quantization, producing an extreme digital effect that may sound glitchy and harsh.

**Crush:** Reduces the bit depth of the waveform to create a classic bit-crushing effect. This adds a gritty, vintage digital sound with pronounced artifacts.

**Blinds 2:** Quantizes the waveform into 2 distinct levels, producing a coarse, binary stepping effect.

**Blinds 4:** Quantizes the waveform into 4 distinct levels, creating a more nuanced stepping effect compared to Blinds 2.

**Blinds 8:** Quantizes the waveform into 8 distinct levels, producing a mild stepped effect.

## Special

**Mirror 1st Half:** Takes the first half of the waveform and mirrors it onto the second half, creating a symmetrical, phase-shifted effect that adds depth and width. This shaper ignores the second half of the original waveform.

**Mirror 2nd Half:** Takes the second half of the waveform and mirrors it onto the first half, creating a symmetrical, phase-shifted effect that adds depth and width. This shaper ignores the first half of the original waveform.

**Mix Mirror 1st Half:** Takes the first half of the waveform and mirrors it onto the second half, creating a symmetrical, phase-shifted effect that adds depth and width. The result is then mixed with the original waveform.

**Mirror 2nd Half:** Takes the second half of the waveform and mirrors it onto the first half, creating a symmetrical, phase-shifted effect that adds depth and width. The result is then mixed with the original waveform.

**Sync 2:** Fits two cycles of the wave shape in the space of one frame, effectively doubling the frequency when played (an octave about)

**Sync 3:** Fits three cycles of the wave shape in the space of one frame.

**Sync 4:** Fits four cycles of the wave shape in the space of one frame.

**Harmonics Soft:** Enhances the waveform by adding subtle harmonic overtones, creating a richer, more complex sound with minimal distortion.

**Harmonics Medium:** Introduces stronger harmonic overtones, giving the waveform a fuller and more textured quality with moderate distortion.

**Harmonics Hard:** Applies significant harmonic enhancement, adding a complex and distorted character to the waveform for a bold, powerful sound.

**Harmonic Warp:** Warps the waveform by applying a harmonic distortion, creating a sine-shaped transformation that enhances harmonic richness, especially on the positive half of the waveform.

**Phase Warp:** Introduces phase warping to the waveform, bending its phase non-linearly to create a complex, phase-shifted effect that adds movement and intrigue.

## Waves Interaction

The Interaction menu selects how Wave 1 and Wave 2 combine or interact across the final wavetable. This feature provides a variety of blending, crossfading, and modulation options to create dynamic and unique sonic textures.

## Interaction Types

### Off

**Off (Wave 1 only):** Uses wave 1 as is and does not use wave 2 in the resulting file.

**Off (Wave 2 only):** Uses wave 2 as is and does not use wave 1 in the resulting file.

### Crossfades

**Crossfade:** Gradually transitions from Wave 1 to Wave 2 across the wavetable, creating a smooth linear blend from one waveform to the other.

**Crossfade Exp:** Applies an exponential curve to the crossfade from Wave 1 to Wave 2, resulting in a faster transition towards Wave 2, adding a more dynamic, accelerating blend.

**Crossfade Log:** Utilizes a logarithmic curve in the crossfade from Wave 1 to Wave 2, creating a slower initial transition that gradually speeds up, offering a unique, decelerating effect for more nuanced blending.

**Fade In 2:** Starts with Wave 1 and progressively mixes in Wave 2 until Wave 2 is fully present at the end, adding Wave 2 gradually.

**Fade Out 2:** Begins with both Wave 1 and Wave 2 fully mixed, then gradually fades out Wave 2, leaving Wave 1 prominent at the end.

**Fade In/Out 2:** Fades in Wave 2 in the first half and fades it out in the second half, creating a dynamic mix that shifts the balance midway through.

**Zigzag Up 2:** Alternates between fading in and fading out Wave 2 across segments, creating a rhythmic, fluctuating effect.

**Zigzag Down 2:** Opposite of Zigzag Up, starting with Wave 2 fading out and ending with it fading in.

**Zig Zag Both:** Alternates between fading Wave 1 out and Wave 2 in, and vice versa, creating an intricate, dynamic interplay between the two waves.

### Mixing

**Mix:** Combines Wave 1 and Wave 2 equally throughout the entire wavetable, producing a mixed sound that includes characteristics of both waves.

**Mix 75/25:** Mixes Wave 1 and Wave 2 in a 75/25 ratio, favoring Wave 1 but adding subtle hints of Wave 2 for texture.

**Mix 65/35:** A mix where Wave 1 and Wave 2 are blended in a 65/35 ratio, providing a balance that still highlights Wave 1 but with more of Wave 2 present than in the 75/25 mix.

**Mix Inverted:** Subtracts Wave 2 from Wave 1, producing a sound that emphasizes the differences between the two waves for a unique effect.

**Mix Inverted 75/25:** A variant of Mix Inverted, with a 75/25 balance between Wave 1 and the inverted version of Wave 2, creating an altered, less balanced mix.

### Modulations

**Ring Modulation 50/50:** Multiplies Wave 1 and Wave 2 to produce complex harmonic content with equal emphasis on both waves.

**Ring Modulation 75/25:** A variation of ring modulation with a 75/25 emphasis, favoring Wave 1 in the modulation.

**Amplitude Modulation 50/50:** Applies amplitude modulation between Wave 1 and Wave 2, producing a modulated, pulsing sound with both waves contributing equally.

**Amplitude Modulation 75/25:** Similar to the 50/50 version but with a stronger presence of Wave 1, creating a more defined, yet modulated sound.

**Morphing Phase Shift:** Modulates the phase of Wave 2 based on a phase shift controlled by Wave 1, resulting in a complex and evolving waveform.

**Phase Mod 1->2:** Uses Wave 1 to modulate the phase of Wave 2, introducing phase-shifted characteristics controlled by Wave 1.

**Phase Mod 2->1:** Reverses the process of Phase Mod 1->2, using Wave 2 to modulate the phase of Wave 1 for a similar but distinct effect.

**Smooth Phase Mod 1->2:** Similar to Phase Mod 1->2 but with smoothed transitions for a gentler phase modulation effect.

**Smooth Phase Mod 2->1:** Applies a smoothed phase modulation where Wave 2 modulates Wave 1, resulting in a smoother, phase-shifted blend.

**Phase Interleaved Mix:** Interleaves segments of Wave 1 and Wave 2 in alternating segments, creating a segmented morph with smooth transitions.

## Special

**Wave Stack:** Creates a rich and dynamic waveform by blending Wave 1 and Wave 2 with additional layers of their reversed versions. This process includes halving the amplitude of the reversed waves to ensure a balanced mix. Ideal for generating harmonically complex and texturally dense waveforms.

**Spectral Morphing:** Transforms the spectral content between Wave 1 and Wave 2, producing a unique effect that shifts the frequency components smoothly.

**Random Cross-Selection:** Randomly selects parts of Wave 1 and Wave 2 for blending, with added smoothing for seamless transitions.

**Add/Sub Mix:** Alternates between adding and subtracting Wave 1 and Wave 2, creating a dynamic and complex waveform.

**Stutter Mix:** Applies a rhythmic stutter effect to Wave 2 during the crossfade, introducing short breaks and creating a rhythmic variation.

**Reverse Cross:** Reverses Wave 2 and blends it into Wave 1 for a unique, backwards crossfade effect with smooth transitions.

**Granular Cross:** Divides each waveform into small grains and randomly pulls grains from Wave 1 or Wave 2 to form the final waveform. This granular approach creates a texture-rich, glitchy crossfade effect that evolves unpredictably.

**Glitch Cross:** Applies a glitch effect during the crossfade by randomly inverting or zeroing out small segments of the waveform. This results in a dynamic crossfade with rhythmic interruptions and "glitches" for a chaotic, digital texture.

**Grain Flow:** Creates a chaotic, granular morph between Wave 1 and Wave 2 by breaking the waveforms into small grains. Each grain is randomly selected from either Wave 1 or Wave 2, and a smooth transition is applied between neighboring grains to produce a smooth and evolving texture.

**Multiband Cross:** Segments the frequency spectrum into multiple bands and applies a crossfade to each band independently. This produces a frequency-split morphing effect, where

different parts of the spectrum transition from Wave 1 to Wave 2 at varying rates, creating a complex, layered sound.

**Multiband Zigzag:** Similar to Multiband Cross but adds a zigzagging effect within each frequency band. Each frequency band alternates between Wave 1 and Wave 2 in a zigzag pattern across multiple segments, resulting in intricate, oscillating transitions within the spectrum.

**Noise Blend:** Blends Wave 1 and Wave 2 with a noise component that varies over time, adding randomness and texture to the crossfade.

**Harmonic Blend:** Applies harmonic weights to Wave 2 and combines it with Wave 1, adding harmonic complexity as the waves morph.

**Even 1 + Odd 2:** Combines the even harmonics from Wave 1 with the odd harmonics from Wave 2, producing a unique hybrid waveform that emphasizes contrasting spectral characteristics.

**Odd 1 + Even 2:** Combines the odd harmonics from Wave 1 with the even harmonics from Wave 2, creating a complementary hybrid waveform with an inverted harmonic emphasis compared to "Even 1 + Odd 2."

## Random

The different parameters for both Wave 1 and Wave 2, as well as the shaper stages interaction can be randomized. This is especially powerful in combination with the Create Random Batch feature. Each parameter has a checkbox next to it, when this checkbox is on the parameter will be randomized - both when clicking the **Random** button or when batch creating wavetables.

- Holding SHIFT while toggling one of the random checkboxes for one of the wave shapes will toggle all of the random checkboxes for that respective wave.
- Holding SHIFT while clicking the random button will toggle all the random checkboxes in the application.

## Wavetable Creation

### Export Settings

Typically wavetable synths each expect a certain combination of frame size and cycle count (there are those who can work with multiple combinations).

The app can export three types of files: .wav, .wt and .h2p. The .wav export will cover the vast majority of synths and modules and will be exported for all export cases. The .wt format is compatible with Synapse Audio Dune 2 and 3. The .h2p format is compatible with U-He Zebra 2 and Zebra HZ.

**Frame Size:** Selects the number of samples per cycle. Different wavetable synths require different sizes. When selecting "All Sizes", a file will be created for every frame size possible (in combination with the Cycles selection). When selecting "Modern", a file will be created for each of 256, 512, 1024, 2048 frame sizes (in combination with the Cycles and Bit Depth selections).

For absolute maximum compatibility select “All Sizes”. For all intents and purposes, selecting “Modern” will cover virtually all use cases. When selecting “Zebra 2”, the frame size is set internally to 128 and a .h2p file is created along with the .wav file. Zebra h2p files will also be created for the “All Sizes” and “Modern” selections. When selecting “Dune”, the frame size is set internally to 2048 and a .wt file is created along with the .wav file. Dune .wt files will also be created for the “All Sizes” and “Modern” selections.

**Cycles:** Selects the number of cycles present in the wavetable. When selecting “All Cycles” a file will be created for every cycle count possible (in combination with the Frame Size selection. When selecting “Modern”, a file will be created for each of 64, 100, 256 cycle counts (in combination with the Frame Size and Bit Depth selections). For absolute maximum compatibility select “All Cycles”. For all intents and purposes, selecting “Modern” will cover virtually all use cases. When “Zebra 2” is selected in the Frame Size menu the cycle count is locked to 16.

**Sample Rate:** Selects the sample rate of the exported .wav file. Note that sample rate has little relevance in the wavetable context, unless a synth needs a specific sample rate, there is no need to go with higher rates.

**Bit Depth:** Selects the bit depth of the exported .wav file. It is recommended to export at 16 bit for maximum compatibility, unless a synth needs a specific bit depth (e.g. Phase Plant requires 32 bit). When “All” is selected, files will be created for all possible bit depths (in combination with the Frame Size and Cycles selections).

## Creation

It is possible to create either a single wavetable file from the current selected parameters, or a random batch of files where each wavetable’s randomized parameters depend on the selected options in the Random section. When creating wavetables, a progress bar will be displayed indicating file progress as a percentage of files created out of the total.

**Folder:** Wavetables are created by default in the user Documents folder, in the Ocean Swift Wavetable Creator\Wavetables subfolder. Selecting a different folder will use the selected folder for export. When tooltips are enabled, hovering over the folder button will display the current selected export folder. When changing the export folder, make sure the application has write permissions to that folder.

**Name:** Entry for the base file name from which the created wavetable files will be created.

**Create Wavetable:** This will create a wavetable file based on the selected settings. When Frame Size is set to process all sizes, wavetable files will be created for each frame size. Similarly, when Cycles is set to process all cycles, wavetable files will be created for each cycle count. When both Frame Size and Cycles are set to all, wavetable files will be created for all possible combinations.

**Create Random Batch:** This will create a batch of wavetables. Each wavetable in the batch will randomize the settings based on which parameter groups are enabled for randomizing. After clicking, a popup window will display asking for a folder name. Leaving this empty will generate a folder name based on a timestamp. Next, an option to select how many batch wavetables to create will come up. When batch creating, it is best practice to ensure all possible frame sizes and cycle counts are created by selecting “All Sizes” in the Frame Size menu and “All Cycles” in the Cycles menu. Each cycle count will generate a subfolder, and each frame size will be

generated in its own subfolder of the relevant cycle count folder. Each Frame Size subfolder will also contain a FileInfo.txt that ensures compatibility with Serum when importing from that folder.

**Singles:** When singles is enabled, exporting wavetables will also create files for every single cycle in the wavetable, and account for any batch operations as well.

**Cancel:** When creating wavetable files, the Cancel button will be shown on screen. Clicking it will stop the file creation operation.

## Player

The audio player provides a preview of what the wavetable will sound like, giving an indication as to the tonal character of the wavetable and flow of the morph.

**Driver:** Selects the sound driver for the audio preview.

**Frequency:** Selects the set frequency for the audio preview.

**Volume:** Sets the volume for the audio preview.

**Play:** Turns the audio preview on or off.

## Visualizer

The visualizer gives a rough overview of how the resulting wavetable will look like with the current selected settings. When Frame Size is set to process all sizes, the visualizer applies the 64 frame size to the visualization. Similarly, when Cycles is set to process all cycles, the visualizer applies the 16 cycles count to the visualization.

## Presets

On first launch, an init.json preset is created in the Presets subfolder of the application files. Overwriting this preset will launch the application with the values saved with this init preset. It is otherwise possible to both save and load presets, retaining the currently selected parameter settings.

**Save Preset:** Saves a preset to file.

**Load Preset:** Loads a preset from file.

## Compatibility

The created wavetables are compatible with a wide range of Wavetable applications. This includes software and hardware synths, Eurorack modules, iOS/Android apps, plugin/instrument development platforms and scientific tools.

The list below gathers some of the known compatible products, if a product does not appear on this list, as long as it can load custom wavetables, with one or more of the available Frame Size and Cycle Count combinations then it is fully compatible.

| Synth or Module             | Frame Size | Cycle Count |
|-----------------------------|------------|-------------|
| 1010 Music Bitbox           | 2048       | Up to 256   |
| 1010 Music Synthbox         | 2048       | Up to 256   |
| Akai - MPC Key 61           | 2048       | 256         |
| ALM046 - MCO (Eurorack)     | 256        | 64          |
| ALM046 - MCO (VCV)          | 256        | 64          |
| Ableton - Wavetable         | 1024       | 64          |
| Arturia - Pigments          | 2048       | 256         |
| AudioMulch                  | Varies     | Varies      |
| AudioKit - Synth One        | 2048       | 64          |
| Axoloti                     | Varies     | Varies      |
| Bespoke Synth               | Varies     | Varies      |
| Bitwig Studio               | 256        | Varies      |
| Caustic 3                   | 2048       | 64          |
| DSP Robotics - Flowstone    | Varies     | Varies      |
| Elektron - Digitone         | 256        | Varies      |
| Expert Sleepers Disting EX  | 256        | Varies      |
| FMOD Studio                 | Varies     | Varies      |
| GNU Octave                  | Varies     | Varies      |
| Groove Synthesis - 3rd Wave | Varies     | Varies      |
| HISE                        | 2048       | 100         |
| IK Multimedia - Syntronik 2 | 512        | 64          |
| JUCE                        | Varies     | Varies      |
| Kilohearts - Phase Plant    | 2048       | 256         |
| Korg - Modwave              | 2048       | 64          |
| KV331 Audio - SynthMaster   | 1024/2048  | 64/256      |
| LabVIEW                     | Any        | Any         |

|                               |           |           |
|-------------------------------|-----------|-----------|
| MATLAB                        | Any       | Any       |
| Max for Live                  | Varies    | Varies    |
| Moog Model 15                 | 1024/2048 | 100       |
| MUX Modular                   | 256-2048  | Varies    |
| Nano Studio 2 (Eden Synth)    | Varies    | Varies    |
| Native Instruments - Kontakt  | 2048      | 100       |
| Native Instruments - Reaktor  | 2048      | 100       |
| Novation - Peak               | 256       | 5         |
| Novation - Summit             | 256       | 5         |
| Ocean Swift - Defiant WT      | 2048      | 100       |
| Ocean Swift - OSS Enterprise  | 2048      | 100       |
| Ocean Swift - Pathfinder WT   | 2048      | 100       |
| Ocean Swift - Porphyra Hybrid | 2048      | 100       |
| Osiris - Wavetable Oscillator | 256/512   | 64        |
| Parawave - Rapid              | 2048      | 256       |
| Piston Honda MkIII            | 256       | Varies    |
| Pure Data                     | Varies    | Varies    |
| Python (SciPy, NumPy)         | Varies    | Varies    |
| Reason - Europa               | 1024      | Varies    |
| Sonic Academy - ANA 2         | 2048      | 256       |
| Sonic Core - Vectron          | 128       | 1         |
| Steinberg - Halion            | 2048      | 100       |
| SunVox                        | 2048      | 100       |
| Supercollider                 | Varies    | Varies    |
| SurgeXT                       | 8-4096    | Up to 512 |
| Symbolic Sound - Kyma         | Varies    | Varies    |
| Synapse Audio - Dune 2        | 2048      | Up to 256 |

|                                  |           |           |
|----------------------------------|-----------|-----------|
| Synapse Audio - Dune 3           | 1024/2048 | Varies    |
| Synth Edit                       | Varies    | Varies    |
| Synthesis Technology - E352      | 256/512   | 64        |
| Synthesis Technology - E370      | 256/512   | 64        |
| Synthesis Technology - Wave Edit | 256       | 64        |
| Tiptop Audio Vortex              | 256       | 64        |
| Tone2 - Icarus                   | 1024-2048 | 64-256    |
| Tracktion - BioTek 2             | 2048      | Varies    |
| Tranzwave (Kontakt)              | 2048      | 100       |
| U-He - Hive 2                    | 2048      | 256       |
| U-He - Zebra 2                   | 128       | 16        |
| U-He - ZebraHZ                   | 128       | 16        |
| U-He Zebralette 3                | 2048      | Varies    |
| UVI - Falcon                     | 2048      | 256       |
| Vital                            | 2048      | 256       |
| VCV Rack - Seven Seas            | 2048      | 100       |
| VPS Avenger 2                    | 2048      | Up to 256 |
| Waldorf - Blofeld                | 128       | Varies    |
| Waldorf - M                      | 128       | Varies    |
| Waldorf - Nave                   | 256       | 128-256   |
| Waldorf - NW1                    | 32-1024   | Varies    |
| Waldorf - Quantum / Iridium      | 2048      | 256       |
| Waves - Codex                    | 512       | 64        |
| Wwise                            | Varies    | Varies    |
| Xfer - Serum                     | 2048      | 256       |
| Ziqal - Dimension                | 2048      | 64        |

# Wavetable Bi-Cycle Morpher

## Introduction

The Ocean Swift Wavetable Bi-Cycle Morpher brings the paradigm of the BiCycle oscillator from the Ocean Swift modular environment into the Sound Creator Suite — and supercharges it. The concept is simple and powerful: load two single-cycle waveforms, and the tool builds an entire wavetable that morphs from the first cycle to the second across its frames. Where the BiCycle module offers a single linear crossfade between its two anchors, the Bi-Cycle Morpher opens up the complete Interaction palette of the Wavetable Creator: crossfades, mixes, ring and amplitude modulation, phase modulation, spectral morphing, granular and glitch treatments, harmonic blends, and more. Any two cycles you can find or make become the raw material for a full, evolving wavetable.

Each anchor can be pre-shaped by up to four serial shaper stages before the morph, and a final main shaper can be applied to the composed table. The result is auditioned with the built-in audio player, displayed in the visualizer, and exported as WAV — with Dune .wt and Zebra 2 .h2p formats supported — using the same flexible frame size, cycle count, and batch machinery as the Wavetable Creator. Combined with randomization that can even pick random cycles from your folders, the Bi-Cycle Morpher turns any collection of single cycles into a wavetable factory.

## Anchors

At the heart of the tool are two anchor slots, Anchor 1 and Anchor 2. Each holds one single-cycle waveform loaded from a WAV file.

Strict cycle length: an anchor file must decode to exactly 2048 samples. This matches the BiCycle module's requirement and guarantees the anchors are true single cycles at the standard resolution. Files of any other length are rejected, and the log reports the actual decoded sample count so you can see why. Stereo files are accepted and downmixed to mono.

Each anchor row offers several ways to load a cycle:

- **Load:** Opens a file dialog to select a WAV file. The dialog remembers the last folder you browsed, including across sessions.
- **File menu:** Once a file is loaded, the name area becomes a dropdown listing every WAV file in that file's folder, in alphabetical order. Picking an entry loads it directly. If an invalid file is picked, a warning is logged and the selection reverts.
- **Arrows:** The ◀ and ▶ buttons step backward and forward through the anchor's folder. Files that are not valid 2048-sample cycles are skipped automatically.

- **Drag and drop:** Drop a WAV file onto an anchor's file menu to load it, or drop an entire folder to scan it and load its first valid cycle. Invalid files are skipped with the same rules as the arrows.

If only one anchor is loaded, the wavetable holds that cycle static across all frames — useful for turning a single cycle into a full-size wavetable file. If no anchors are loaded, there is nothing to build and creation is refused with a message in the log.

## Shapers

Each anchor has four serial shaper stages — Shaper A, B, C, and D — applied in order to that anchor before the morph. A final Shaper Main stage is applied to the fully composed wavetable after the interaction. Every stage defaults to None, which is a true bypass: only the stages you set do anything.

The shaper palette is identical to the Wavetable Creator's; see the Wave Shapers section of the Wavetable Creator manual for descriptions of every processor. Because the four per-anchor stages run in series, the same processors in a different order can produce very different results — stacking and reordering the stages is part of the fun.

## Interaction

The Interaction menu selects how Anchor 1 and Anchor 2 combine across the wavetable. This is the same interaction system as the Wavetable Creator — see the Waves Interaction section of that manual for descriptions of every mode. A few notes specific to this tool:

Crossfade is the exact linear morph performed by the BiCycle module: the table transitions evenly from Anchor 1 at the first frame to Anchor 2 at the last. It is the default and the natural starting point. Off (Wave 1 Only) and Off (Wave 2 Only) fill the table with just one anchor, ignoring the other. Every other mode — the exponential and zigzag crossfades, the mixes, the modulations, the spectral and granular treatments — replaces BiCycle's simple line with something the module never had.

The morph preserves the natural level relationships between frames. If the two anchors partially cancel in the middle of a crossfade, the middle frames get quieter, exactly as they do in the BiCycle module — the tool does not artificially boost quiet frames back to full level. The finished table is normalized as a whole, so the loudest point of the morph sits at full scale and everything else keeps its proportion.

## Random

The randomization system covers the shapers, the interaction, and — uniquely in this tool — the anchor files themselves. Each element has a checkbox: when it is on, that element is randomized by the Random button and by batch creation; when it is off, it keeps its current value. The checkboxes control randomization only; to disable a shaper, set it to None.

The checkbox next to each anchor's file menu enables file randomization for that anchor: randomizing will pick a random valid cycle from that anchor's scanned folder. Load one file from a folder of single cycles, enable the checkbox, and every randomize or batch item can start from a different cycle. With both anchors pointed at folders and both checkboxes on, a random batch generates morphs between random pairs of cycles — an extremely fast way to explore a cycle library.

Holding SHIFT while clicking the Random button toggles all of the randomization checkboxes in the tool at once.

Randomization is seed-based: every randomized result is reproducible, and the seed is stored with presets, so what you audition is exactly what you export and exactly what a reloaded preset regenerates — including the stochastic interaction modes like Granular Cross and Noise Blend.

## Wavetable Creation

### Export Settings

The export system is the same as the Wavetable Creator's. The app exports .wav files in all cases, .wt files compatible with Synapse Audio Dune, and .h2p files compatible with U-He Zebra 2.

- **Frame Size:** Selects the number of samples per cycle in the exported files (64 through 4096, default 2048). The loaded anchors are always 2048 samples; they are resampled to the target frame size when the table is built. "All Sizes" and "Modern" create files for multiple sizes, and the "Zebra 2" and "Dune" selections create the respective synth formats alongside the WAV — all exactly as described in the Wavetable Creator manual.
- **Cycles:** Selects the number of frames in the wavetable (5 through 512, default 100 — the BiCycle module's native frame count). "All Cycles" and "Modern" create files for multiple counts. When "Zebra 2" is selected in the Frame Size menu, the cycle count is locked to 16.
- **Sample Rate:** Selects the sample rate of the exported WAV files.
- **Bit Depth:** Selects the bit depth (8, 16, or 32, with "All" creating every depth). 16 bit is recommended for maximum compatibility.
- **Singles:** When enabled, every individual cycle of the wavetable is also exported as its own file.

### Creation

- **Folder:** Wavetables are created by default in the Wavetable Bi-Cycle Morpher\Wavetables subfolder of the application documents folder. The Folder button selects a different export folder; hovering over it shows the current selection.
- **Name:** The base file name for the created files. Create Wavetable: Creates wavetable file(s) from the current settings. A progress bar replaces the visualizer during creation, and the Cancel button stops the operation.

- **Batch:** Creates a batch of randomized wavetables. A popup asks for a folder name (leave empty for a timestamped name), then for the batch size (5 to 100). Each item randomizes according to the Random checkboxes — including random anchor files from their folders when enabled — and the batch is organized into Cycles and Frame Size subfolders with a FolderInfo.txt for Serum compatibility, matching the Wavetable Creator's batch layout. Every batch item is seed-reproducible.

## Player

The audio player previews the wavetable, indicating the tonal character and the flow of the morph. It is the same player as the Wavetable Creator's.

- **Frequency:** Selects the playback frequency of the preview oscillator.
- **Volume:** Sets the preview volume.
- **Position:** Scrubs through the wavetable's cycles manually. A marker line on the visualizer tracks the scrub position, so you can see exactly which part of the morph you are hearing.
- **Play:** Turns the audio preview on or off.

## Visualizer

The visualizer shows the full composed wavetable — all cycles side by side, left to right — and updates live as you change anchors, shapers, or the interaction. During file creation it is temporarily replaced by a progress bar. The dashed marker line indicates the current player scrub position.

## Presets

On first launch, an init.json preset is created in the Presets subfolder of the application files, and the tool restores its last state from it on every launch. Presets store everything: the anchor file paths, all shaper and interaction settings, the export configuration, and the random seed. Loading a preset reloads the anchor files from their saved paths; if a file has moved or been deleted, a warning is logged and that anchor is left empty.

- **Save Preset:** Saves the current settings to a preset file.
- **Load Preset:** Loads settings from a preset file. **Reset:** Returns all settings to their defaults and clears both anchors.

# Wavetable Cycle Creator

## Introduction

The Wavetable Cycle Creator is a focused, intuitive tool for musicians, producers, and sound designers who need to craft single-cycle waveforms quickly and precisely. Where a full wavetable evolves across many frames, a single-cycle waveform is one perfectly-formed period of sound — the fundamental building block loaded into wavetable synthesizers, samplers, and single-cycle oscillators. The Wavetable Cycle Creator gives you a large palette of generators and shaping processors to sculpt exactly the cycle you want, see it drawn in real time, and export it as a ready-to-use WAV file.

The application shares its sound-generating heart with the Ocean Swift Wavetable Creator, offering the same rich selection of wave shapes — from classic sine, triangle, square, and sawtooth shapes through additive, FM, harmonic, and noise-based waveforms — but streamlines the workflow around the single-cycle use case. There is one source waveform rather than two morphing layers, and instead of an interaction stage there are four serial shaping processors that let you stack transformations to taste.

A core strength of the Wavetable Cycle Creator is speed of exploration. Every parameter can be randomized, and a randomized batch export can generate a whole folder of unique single cycles in one action — an efficient way to build a library of raw material to audition and pull from. Each generated file is written with an embedded loop point so that single-cycle-aware instruments recognize it correctly.

Whether you are designing a precise oscillator shape by hand or generating dozens of candidates to sift through, the Wavetable Cycle Creator aims to make single-cycle creation fast, deep, and repeatable. The straightforward interface keeps it approachable, while the depth of the generator and shaper palette ensures there is plenty of territory to explore.

## Source

At the core of the Wavetable Cycle Creator is a single source waveform. Unlike the Wavetable Creator, which morphs between five keyframes across the length of a wavetable, the Cycle Creator generates one waveform shape as the basis for the single cycle. That source can then be shaped by up to four processing stages before export.

The source is selected from the Source menu. Each source is peak-normalized so that it exports at a consistent level regardless of the shape chosen.

## Wave Shapes

The Wavetable Cycle Creator offers a wide selection of wave shapes that serve as the starting point for your single cycle.

## Wave Shapers

The generated source can be further modulated, shaped, and manipulated by up to four shaper stages: Shaper A, Shaper B, Shaper C, and Shaper D. The stages are applied in series — the source passes through A, then B, then C, then D — so the order matters, and effects can be stacked to build up complex transformations from simple processors.

Each shaper stage is set to None by default. A stage set to None is a true bypass: it leaves the waveform completely untouched, with no change to level or shape. This makes it easy to use only as many stages as you need — set the ones you want and leave the rest at None. Selecting any processor other than None engages that stage.

Note on ordering: because the four stages run in series, the same two processors can produce different results depending on which stage each occupies. For example, filtering before distorting sounds different from distorting before filtering. Experimenting with the stage order is part of the creative process.

## Random

The source and each of the four shaper stages can be randomized. Each has a checkbox next to it: when the checkbox is on, that parameter will be randomized — both when clicking the Randomize button and when creating a randomized batch. When a checkbox is off, that parameter keeps its current value and is left out of randomization. The checkboxes control randomization only; they do not disable a shaper. To disable a shaper, set its menu to None.

Holding SHIFT while clicking the Randomize button toggles all of the randomization checkboxes in the tool at once, making it easy to flip between randomizing everything and randomizing nothing.

Randomization is seed-based, which means every randomized result is reproducible. When a cycle is randomized, its underlying random seed is stored along with the rest of its settings. Saving a preset, using undo/redo, or re-exporting therefore reproduces the exact same waveform — what you see is what you get, and what you export is exactly what you previewed.

## Single Cycle Creation

### Export Settings

Single-cycle files are exported as WAV, which is compatible with the vast majority of wavetable synths, samplers, and single-cycle oscillators. Each exported file includes an embedded loop point spanning the whole cycle, so instruments that read loop metadata will treat the file correctly as a single-cycle waveform.

- **Frame Size:** Selects the number of samples in the exported cycle. The available sizes are 128, 256, 512, 1024, 2048, and 4096, with 2048 as the default. Different instruments

prefer different frame sizes; 2048 is a widely-compatible modern choice, while smaller sizes suit lo-fi, chiptune, or hardware targets that expect shorter cycles.

- **Sample Rate:** Selects the sample rate written into the exported WAV header (44100, 48000, 88200, or 96000). Sample rate has little relevance in the single-cycle context; unless an instrument requires a specific rate, there is no need to use higher rates.
- **Bit Depth:** Selects the bit depth of the exported WAV file (8, 16, or 32). Exporting at 16 bit is recommended for maximum compatibility, unless an instrument requires a specific bit depth (for example, some hosts expect 32-bit float).

## Creation

You can create either a single file from the current settings, or a randomized batch of files whose parameters follow the Random section. When exporting, a progress bar replaces the visualizer to indicate progress as a percentage of files completed.

- **Folder:** Single cycles are created by default in the user Documents folder, in the Ocean Swift Wavetable Cycle Creator\Wavetable Cycles subfolder. Selecting a different folder will use that folder for export. When tooltips are enabled, hovering over the Folder button displays the current export folder. When changing the export folder, make sure the application has write permission to that folder.
- **Name:** The base file name from which exported files are named. A single export is named using the base name, the source shape, the frame size, and the bit depth. A batch export names each file with the base name, a sequential index, the frame size, and the bit depth.
- **Export:** Creates a single WAV file from the current settings. The exported file is exactly the cycle shown in the visualizer. Batch...: Creates a batch of randomized single cycles. After clicking, a popup asks for a folder name; leaving it empty generates a folder name based on a timestamp. A second popup asks how many cycles to create, with choices of 5, 10, 25, 50, 100, 250, or 500. Each cycle in the batch randomizes its parameters according to which checkboxes are enabled in the Random section, and every cycle is written to the chosen folder. Because randomization is seed-based, each file in the batch is individually reproducible.
- **Cancel:** While files are being created, the Cancel button is shown. Clicking it stops the export operation.

## Visualizer

The visualizer displays the current single-cycle waveform in real time, updating as you change the source, the shapers, or the frame size. It shows exactly the cycle that will be exported, so it doubles as an accurate preview. During export, the visualizer area is temporarily replaced by a progress bar, and returns to the waveform display when the operation completes. The Wavetable Cycle Creator is a visual tool and does not include an audio player.

## Presets

On first launch, an init.json preset is created in the Presets subfolder of the application files. Overwriting this preset will launch the application with the values saved in it. It is also possible to save and load presets to retain the current parameter settings, including the random seed so that any randomized cycle is preserved exactly.

- **Save Preset:** Saves the current settings to a preset file.
- **Load Preset:** Loads settings from a preset file. **Reset:** Returns all settings to their defaults.

## Wavetable Extractor

### Introduction

The Wavetable Extractor uses spectral information from a loaded audio file to construct a wavetable. Unlike standard file converters that blindly slice audio into equal chunks, the Wavetable Extractor is a frequency-domain authoring tool. It is designed to intelligently "listen" to your audio, mathematically extract the most stable harmonic cycles, perfectly align their phases, and curate a 100-frame synthesizer wavetable.

### Source Audio

Start by loading the raw audio you want to convert.

- **Select Folder:** Click to choose a directory on your hard drive. The tool will automatically scan and load all valid audio files (.wav, .aif, .aiff) found within it.
- **Target File:** Use the dropdown menu or the < and > arrow buttons to select the specific file you want to process from the loaded folder.
- **Recent:** Quickly re-load previously visited workspace folders.
- **File Specifications:** The engine accepts files up to **3 minutes long** and sampling rates up to **192 kHz**. *Note: The engine will internally downsample the analysis buffer to optimize memory and keep processing blindingly fast without losing any harmonic detail.*

### Extraction Pipeline Settings

This is the core DSP engine. These settings define how the tool selects, shapes, and arranges the 100 frames of your final wavetable.

## Core Analysis

- **Region:** Tells the engine where to look for usable audio.
  - *Auto (Stable Middle):* Automatically finds the most pitch-stable, sustained portion of the sample. Best for standard instruments and synths.
  - *Attack Focus:* Forces the engine to extract frames right from the start of the audio file. Great for punchy plucks and percussive transients.
  - *Full File:* Scans the entire file linearly.
- **Morph Order:** Determines how the extracted frames are arranged into the final 100-frame table.
  - *Sequential (Dynamic Keyframes):* **(Recommended)** Warps the time axis based on timbral changes. It skips boring "dead zones" in the audio and focuses the wavetable entirely on the most dynamic, shifting parts of the sound.
  - *Sequential (Linear Time):* Steps chronologically through the audio file exactly as it was recorded.
  - *Sorted by Brightness:* Ignores time completely and arranges the frames from the darkest (lowest harmonic content) to the brightest. Perfect for creating natural filter-sweep wavetables.

## Spectral Shaping

- **Phase Mode:** Defines how the physical waveform is mathematically reconstructed.
  - *Acoustic (Fundamental Lock):* Perfectly locks the root pitch to prevent morph-cancellation, but preserves the organic, natural shape of the waveform.
  - *Zero-Phase (Synthetic):* Forces all harmonic energy into a single laser-sharp peak. Best for aggressive, synthetic EDM basslines.
  - *Random Phase (Airy):* Scrambles the phase of the upper harmonics. Turns any sound into a breathy, chaotic, whispered pad texture.
- **Character:** Applies macro-level spectral formatting.
  - *Raw:* Leaves the extracted FFT bins untouched.
  - *Clean:* Rolls off extreme, harsh high frequencies.
  - *Warm:* Boosts the low-mid body harmonics and tames the highs.
  - *Dirty:* Injects subtle analog grit and phase-randomization into the upper harmonics.
- **Symmetry:** Magically forces acoustic sounds into classic analog waveform shapes.
  - *Keep All:* Leaves harmonics as they are.
  - *Isolate Odd (Square-like):* Mutes even harmonics, forcing the sound into a hollow, woody, square-wave tone.
  - *Isolate Even (Saw-like):* Mutes odd harmonics for a buzzy, bright texture.

## Detail Sliders (0 - 100%)

- **Smoothness:** Controls the interpolation between frames. Lower values (Linear) give a stepping, robotic feel. Higher values (Cubic) melt the frames together for a liquid sweep.
- **Brightness:** Applies a smooth spectral tilt to darken or brighten the overall wavetable.

- **Harmonic Focus:** Acts like a highly surgical noise-gate. Higher values strip away atonal background noise and hum, leaving only the pure, ringing musical overtones.
- **Warp (Inharmonic):** Mathematically stretches the harmonic spacing. Pushing this up turns standard musical notes into clanging, dissonant, metallic bells and gongs.
- **Sub Inject:** Mathematically injects a massive, pure fundamental sine wave (and 1st octave) into the wavetable. Use this to guarantee deep, room-shaking bass weight on thin Foley or acoustic samples.

## Presets

- **Save / Load Preset:** Save your exact DSP configuration to a `.json` file in the tool's `Presets` folder so you can instantly recall your favorite extraction formulas later.

## Actions

- **Analyze for Preview:** Runs the audio through the DSP engine and loads the resulting 100 frames into memory *without* saving to disk. Use this to rapidly audition and tweak your settings.
- **Extract & Export:** Runs the pipeline and prompts you for a save location. It automatically generates two perfectly formatted files:
  - `filename_16b.wav` (16-bit PCM for legacy synths)
  - `filename_32f.wav` (32-bit Float for modern flagship synths like Serum, Vital, and Oscillays).
- **Cancel:** Instantly aborts the background processing if a file is taking too long.

## Wavetable Preview

Once you hit "Analyze", you can play the wavetable directly in the app to hear exactly how it will behave in a synthesizer. The playback engine uses anti-zipper slew filters, meaning parameter sweeps are perfectly smooth and click-free.

- **Play / Stop:** Toggles the audio stream (with a safe 50ms fade-in/out).
- **Freq:** Changes the playback pitch (55 Hz up to 880 Hz).
- **Morph Slider:** Sweeps through the 100 generated frames. Drag this back and forth to hear the timbral journey of your new wavetable!
- **Vol Slider:** Controls the preview output volume (equipped with an internal safety limiter to protect your speakers).

# Wavetable Viewer

## Introduction

The Wavetable Viewer is a diagnostic tool for inspecting and visualizing .wav based wavetables. It parses continuous audio arrays into individual single-cycle frames and renders them natively using highly optimized 2D and 3D visualizers. The rendering engine utilizes advanced geometry caching and sub-frame positional interpolation for a butter-smooth, 60FPS geometric wireframe aesthetic.

## File Navigation & Parsing

- **Folder / Target File:** Click "Select Folder" or drag and drop a folder directly into the path field to target a directory and automatically populate the dropdown with all available .wav files. Use the "Recent" menu to quickly recall your previously accessed working directories.
- **< / > Arrows:** Instantly cycle to the previous or next file in the directory without opening the dropdown.
- **Frame Size:** Define the exact number of samples that constitute a single wave cycle (from 64 up to 8192 samples). The engine uses this value to mathematically slice the audio file into sequential frames.
- **Safety Limits:** To ensure the application remains perfectly responsive, the engine will reject files larger than 10MB. If a parsed file yields more than 256 individual frames, the display will automatically truncate to show only the first 256 cycles.

## Display Modes

The rendering engine provides four distinct ways to analyze the phase and harmonic evolution of your wavetable. You can customize the visual aesthetic at any time by clicking the Color swatches at the bottom of the interface.

- **3D Parallax:** Renders the frames from back to front with an angled perspective shift. Dedicated Px X and Px Y sliders allow you to manually adjust the horizontal and vertical parallax angles from -50% to 50%.
- **3D Waterfall:** Renders the frames in a straight, vertically stacked layout (Y-axis = time), making it highly effective for spotting harmonic morphing. This mode features a secondary dropdown with four unique aesthetic styles: Hologram (Z-depth fading), Neon (dual-color depth interpolation), Scanner (amplitude-based wave height gradients), and Flow (a fully animated, continuous 60FPS scrolling cascade).
- **2D Mode - Grid:** Lays out all parsed frames simultaneously in a flat, spreadsheet-style 2D grid. In Grid mode, a checkbox allows showing or hiding the grid lines, and an adjacent color swatch appears to let you customize the grid line color.
- **2D Mode - Single:** Isolates a single 2D frame. A horizontal slider appears, allowing you to manually scrub forward and backward through the wavetable index frame by frame.

## Presets

**Save / Load:** Stores your current workspace state - including the targeted folder path, the specific file, the frame size, all customized color swatches, parallax slider values, waterfall styles, and your active display mode - to a JSON file for immediate recall.

## Zeta Ocean Quiz

### Introduction

The Zeta Ocean Quiz is a lightweight, built-in trivia mini-game designed specifically for audio engineers, producers, and synthesizer enthusiasts. It features a curated bank of over 250 questions spanning electronic music history, acoustic physics, DSP math, and hardware trivia.

### How to Play

1. **Read the Prompt:** Upon loading the tool, a random trivia question will be displayed in the center of the screen.
2. **Review Options:** Below the question, you will see four possible answers arranged in a neat, centralized list.
3. **Make Your Choice:** Click the **"Choose"** button next to the answer you believe is correct.
4. **Check the Log:** The main application console log at the bottom of your screen will instantly tell you if your answer was correct or incorrect (and reveal the right answer if you missed it!).
5. **Keep Going:** Regardless of whether you were right or wrong, a new random question will immediately load so you can keep playing.

### Scoring System

The quiz utilizes a "Sudden Death" streak system to test your true knowledge.

- **Current Score:** You start with a score of 0. Every correct answer adds 1 point to your current score.
- **The Penalty:** If you select an incorrect answer, your current streak is broken, and your Current Score instantly resets to 0.
- **High Score Tracking:** The tool constantly monitors your longest streak. If your Current Score surpasses your previous best, the **High Score** counter will update in real-time.

## Zeta Xaxixot

## Introduction

Zeta Xaxixot is a fast-paced territory-capture arcade game. Your goal is to capture **85%** of the ocean area to clear the level. You start with 3 lives and 1 bomb. Lose all your lives, and it's Game Over!

## Controls

- **Arrow Keys:** Move your ship.
- **SPACEBAR:** Detonate a Bomb.
- **ENTER / RETURN:** Pause or Resume.

## The Grid & Hazards

- **Safe Zone (Blue):** Captured area. Move freely, but beware of bouncing anomalies.
- **Open Ocean (Dark) & Trail (Neon Green):** Steer into the ocean to draw a trail.  
**Warning:** If an anomaly touches your trail mid-slice, your ship is destroyed.
- **Anomalies (Red Dots):** Hostile bouncing entities. Contact is lethal.
- **Mines (Pink/Red Blocks):** Procedurally generated solid walls. Contact is lethal.

## Gameplay Mechanics

- **Capturing Territory:** Draw a trail through the Open Ocean and reconnect to the Safe Zone. The smaller of the two newly divided sections will be captured.
- **Defusing Mines:** You cannot crash into mines. To defuse them, capture the ocean area completely surrounding them for a massive point payout.
- **Trapping Anomalies:** Enclose anomalies inside a newly captured area to eliminate them. This grants a score multiplier and **+1 Extra Life per trapped anomaly**.
- **Bombs (Panic Button):** Press **SPACE** to trigger a shockwave. This vaporizes nearby anomalies (+1 Life per kill), defuses nearby mines, and instantly forces a successful capture if you are mid-slice.

## Pickups

Fly over loot to collect it. *Note: Capturing an area containing a pickup without physically grabbing it will destroy the item.*

-  **Neon Green Square:** Instant Score Bonus.
-  **Dark Orange Circle:** +1 Bomb (Maximum: 3).
-  **Bright Blue Circle:** +1 Extra Life (Maximum: 10).

## Interface & Settings

- **Speed Multiplier:** Adjust game speed from **0.50x** to **1.50x**. All point payouts (captures, defusals, kills) scale directly with this multiplier.
- **Audio Controls:** Toggle the looping background music ON/OFF and adjust the volume slider.
- **Restart:** Instantly wipes your current run and starts a fresh game at Level 1.
- **Save / Load Data:** Save your High Score, Speed, and Audio preferences to a **.json** file to carry your progress across sessions.

## Zeta Xong

### Introduction

Zeta Xong is a fast-paced, vertical homage to the classic arcade game of table tennis. Face off against the AI in a neon-drenched arena. The first player to reach 10 points wins the match!

### Controls

- **Left / Right Arrow Keys:** Move your paddle horizontally.
- **ENTER / RETURN:** Pause or Resume the match.

### Interface & Settings

- **Start / Pause:** Begin the match or freeze the action.
- **Audio Controls:** Toggle the looping background music ON/OFF and adjust the master volume slider. *Note: Music only plays while a match is actively running or serving.*
- **Restart:** Instantly wipe the current scoreboard to **0-0** and start a fresh match.

## Change Log

### Version 1.10.x

**FIXED** Dozens of bug fixes across all tools

**ADDED** Last Tool button allowing quick switching between two tools

**ADDED TOOL** MIDI XY Bowl

**ADDED TOOL** OSC XY Bowl

**ADDED TOOL** Wavetable Cycle Creator

**ADDED TOOL** Wavetable Bi-Cycle Morpher

**ADDED TOOL** GUI Sheet Slicer

**IMPROVED TOOL** Knob Maker additional shapes and options

## Version 1.2.0

**ADDED TOOL** Kontakt NKS2 Mapper

**ADDED TOOL** Kontakt Lua Center

**ADDED TOOL** Toolbox Smart Renamer

**IMPROVED TOOL** Knob Maker additional presets

## Version 1.0.0

**ADDED** Initial public release

## Version 0.13.21

**ADDED** Initial alpha release

## Attribution

This software uses the QtAwesome library (MIT License).

Icons are provided by Font Awesome (SIL OFL), Material Design Icons (Apache 2.0), Phosphor (MIT), Remix Icon (Apache 2.0), and Microsoft Codicons (CC BY 4.0).