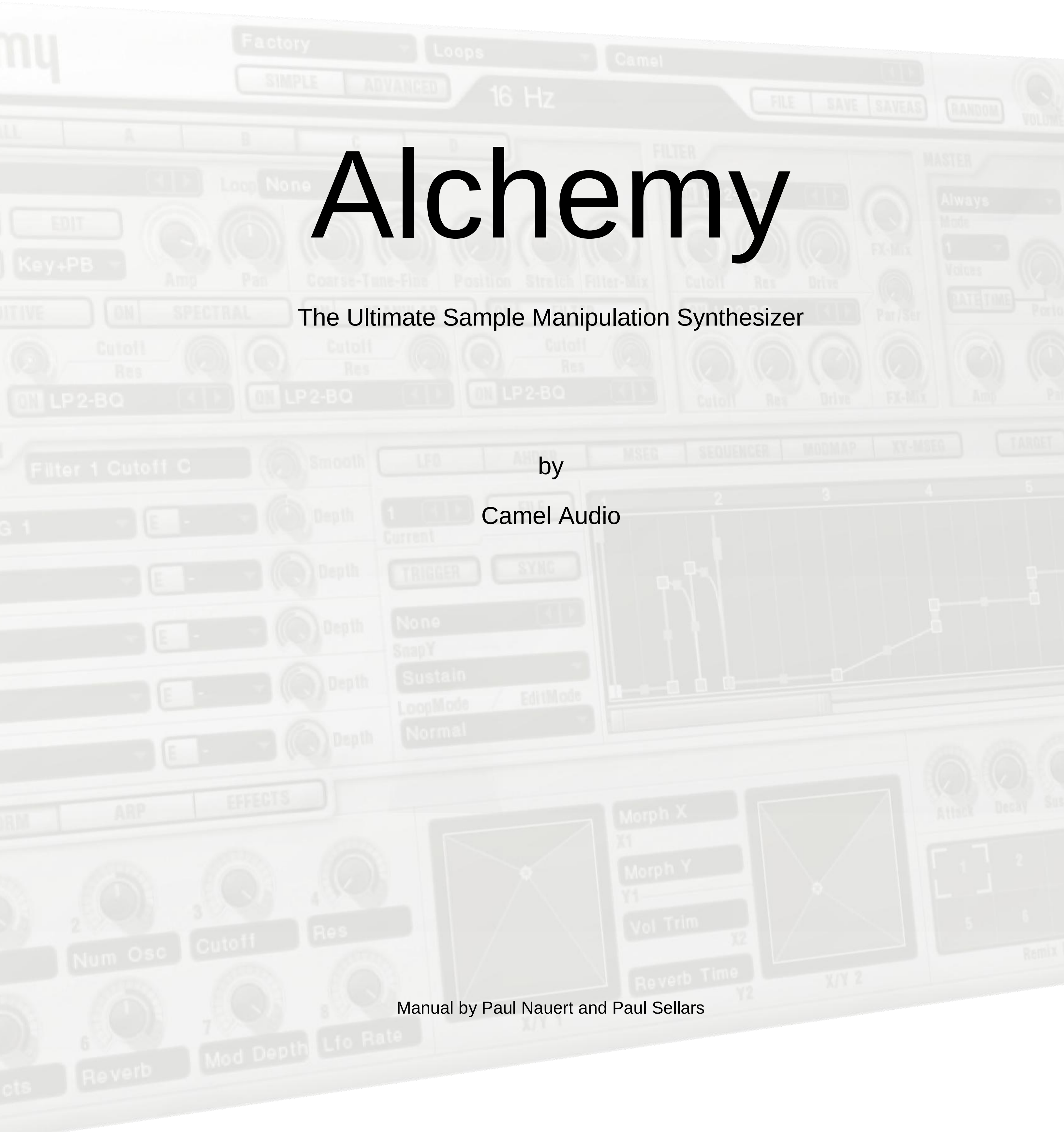


Alchemy

The Ultimate Sample Manipulation Synthesizer

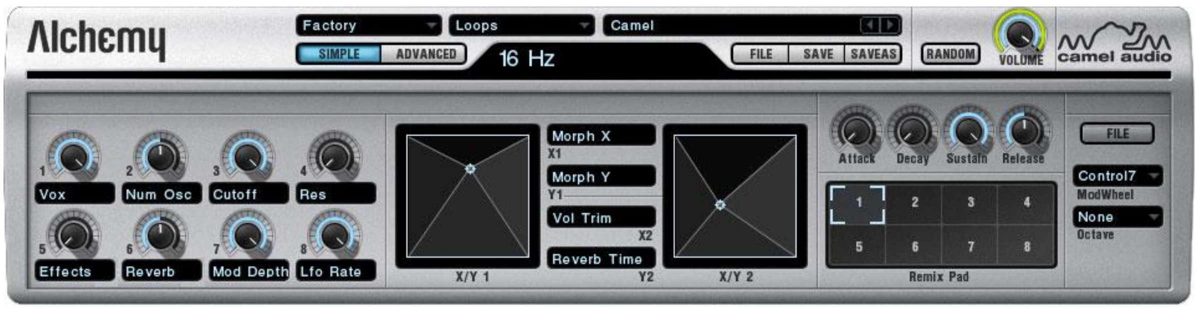
by

Camel Audio

Manual by Paul Nauert and Paul Sellars

Welcome To Alchemy

Alchemy is the ultimate sample manipulation synthesizer. It's a synth powerhouse and yet is very easy to use thanks to its performance controls and remix pads. Just tweak the library of excellent presets from many of the world's top sound designers or dive in and analyse your own samples - it's up to you!



(The SIMPLE interface.)

Alchemy features additive, spectral and granular synthesis and resynthesis, sampling, and a very capable virtual analog engine with unison and PWM. You can morph or crossfade between sources. You can import your own samples from SFZ, WAV or AIFF files. A wide range of analog modelled filters is included, in addition to a flexible rack of effects which includes all those from CamelPhat and CamelSpace as well as many new effects such as a high quality reverb. The innovative modulation system is extremely flexible, yet easy to use. Alchemy also features a powerful arpeggiator with the ability to import the groove from any MIDI file for immediate synchronization to a beat.

(The ADVANCED interface.)

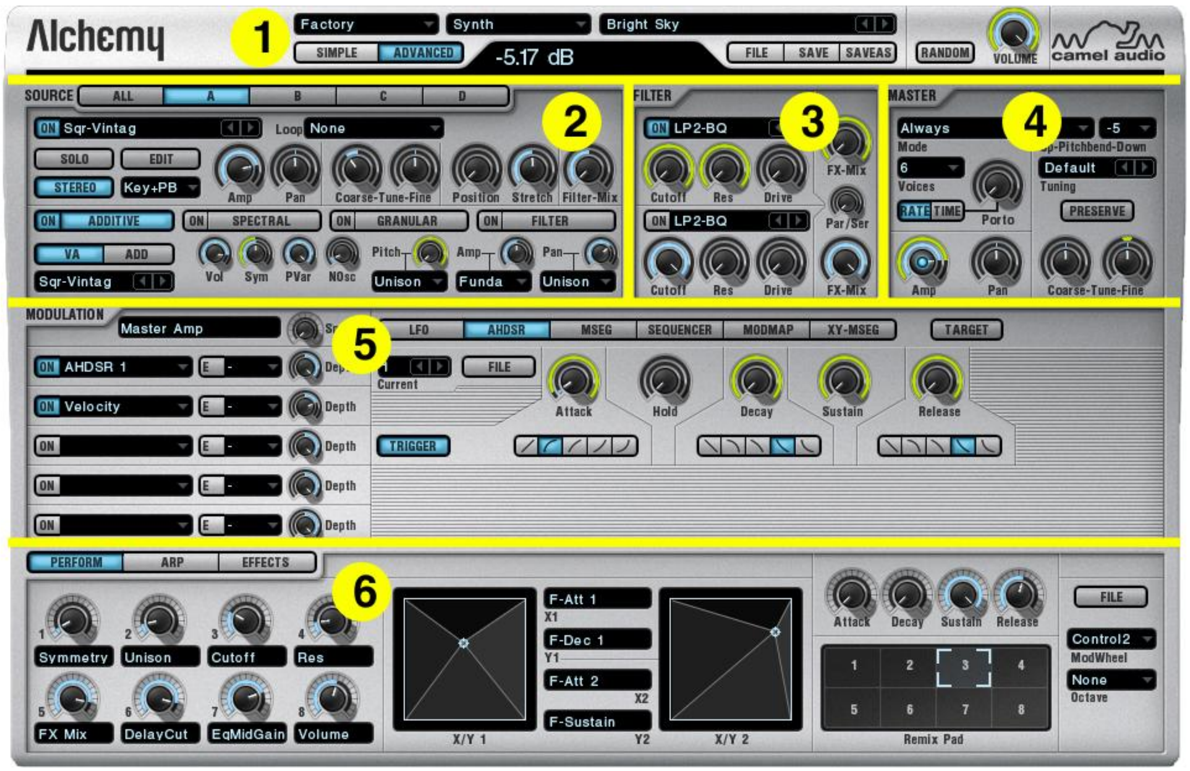
Alchemy ships with over 2GB of samples and analysed content from in-house designers Tim Conrardy and Biomechanoid, as well as designers such as Ian Boddy, Robert Rich, Scott Solida and Nucleus SoundLab. A library of 300 excellent presets from many of the world's top sound designers is included, arranged into categories for rapid access to the sound you require.

Using the Manual

A list of topics is available in the right-hand sidebar of each page. Click a topic to go directly to its page. You can also click on links found in the main text.

If this is your first time browsing through the manual, we suggest you begin with the Overview page. Or dive right in to whatever topic interests you most!

Overview



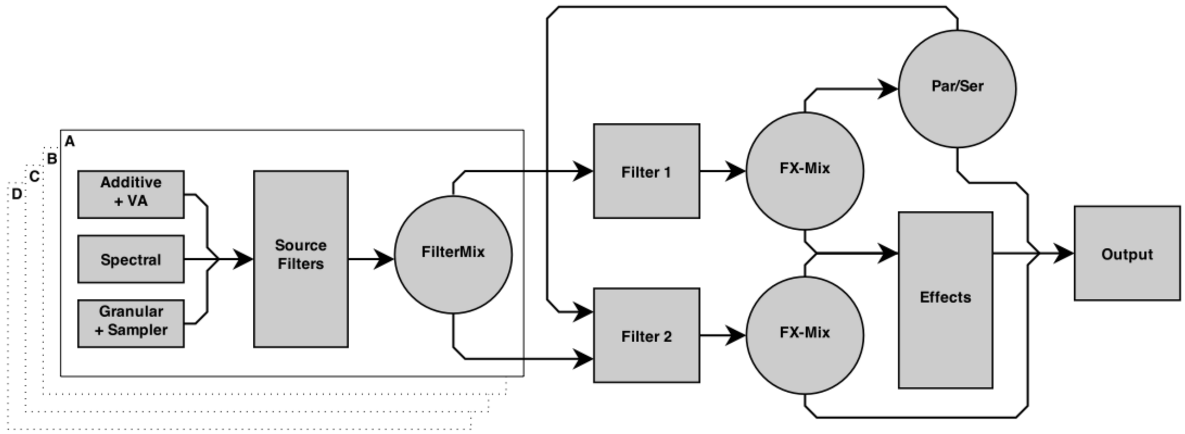
The main or ‘global’ page in Alchemy is divided into six sections, shown above.

1. The Title bar.
2. The Source section.
3. The Filter section.
4. The Master section.
5. The Modulation section.
6. The Perform/Arp/Effects section.

Signal flow in Alchemy

Before you can design new sounds with Alchemy you’ll need to understand how its different parts fit together, and how each of them in turn contributes to what you hear.

At first glance Alchemy may seem complicated, but its layout is relatively simple:



This diagram represents signal flow in Alchemy, reading from left to right, through the different sound-generating and -processing modules.

There are three basic stages:

1. There are four Sources (A, B, C, and D), each consisting of Additive, Spectral, and Granular elements and a bank of Source Filters.
2. There are two main Filters (1 and 2), which can be operated in parallel or in series.
3. After the individual voices mixed together, they pass through the Effects stage.

Installation and requirements

Requirements

To use the Alchemy software you need a computer with at least the following specifications:

Mac
OSX 10.4.9, 2 GHz Intel or G5, 1 GB RAM, 3 GB disk space, Audio Units or VST 2.4 host.
Intel Core2Duo and 2 GB RAM recommended.

Windows
XP SP2 or Vista, 3 GHz Pentium, 1 GB RAM, 3 GB disk space, VST 2.4 host.
Intel Core2Duo and 2 GB RAM recommended.

On a 2GHz Core2Duo the most CPU intensive factory preset uses approximately 50% of one processor when playing six notes. For optimum results, a fast, modern CPU and additional RAM are recommended. For tips on how to conserve processor power, see the ***Troubleshooting** page.*

Installing Alchemy

In order to run Alchemy you must download a number of files and then extract and run the installer. The following files must be downloaded to your desktop:

- Your keyfile
- The factory samples (a .rar file that will be unpacked by the installer)
- The factory presets (a .rar file that will be unpacked by the installer)
- The installer for your chosen platform

1. Go to the Camel Audio web site at http://www.camelaudio.com
2. Log into your user account by clicking on the ‘Log in / create user account’ link at the top right-hand corner of the page and entering your email address and password.
3. Click on the ‘Downloads’ link in the Support Menu on the left of the page.
4. Download the Alchemy keyfile to your desktop, by clicking on the Keyfile link towards the top of the page and selecting the desktop in your browser. If your browser does not allow

you to select the desktop, move this and the other files once they have all been downloaded.

5. Download Alchemy Samples to your desktop

6. Download Alchemy Presets to your desktop

7. Download Alchemy Plugin to your desktop, selecting either the Win or OSX version depending on the computer you are installing to.

8. ***Windows:*** Double click on AlchemyWin-1-xx.zip on your desktop to extract the installer, and then double click on the installer to run it and follow the on-screen instructions. ***Mac:*** Double click on AlchemyMac-1-xx.pkg and follow the on-screen instructions.

9. Start your sequencer application. The keyfile that you saved to your desktop will automatically be moved into the Alchemy installation folder.

Installation is complete. It is now safe to move all installation files from your desktop to a suitable backup location as they are no longer required by Alchemy.

Installing Add-on Sound Banks

1. Go to the Camel Audio web site at http://www.camelaudio.com

2. Log into your user account by clicking on the 'Login / create user account' link at the top right-hand corner of the page and entering your email address and password.

3. Click on the 'Downloads' link in the Support Menu on the left of the page

4. If you paid for the sound bank that you are installing, then you will need to click on the download link for the corresponding keyfile and save that to the desktop.

5. Click on the download link for the add-on sound bank that you wish to install.

6. Select the desktop as the location to which the file is downloaded.

7. ***Windows:*** Click on the Programs->Camel Audio->Alchemy->Install Add-ons option in the Windows Start menu. ***Mac:*** Double-click on Alchemy Sound Bank Installer in the /Applications/Alchemy folder.

8. Start your sequencer application. If you placed a keyfile on your desktop then it will automatically be moved into the Alchemy plugin folder.

Installation is complete. It is now safe to move all installation files from your desktop to another location as they are no longer required by Alchemy.

Uninstalling

Windows

Click on the Programs->Camel Audio->Alchemy->Uninstall option in the Windows Start menu.

Mac

1. Delete files:
   /Library/Audio/Plug-Ins/Components/Alchemy.component
   /Library/Audio/Plug-Ins/VST/Alchemy.vst.

2. Delete folder:
   /Library/Application Support/Camel Audio/Alchemy

   Note that all of the contents of the Alchemy data folder will be deleted, including any user presets and samples that have been saved there.

Upgrading

To install a new version of Alchemy, just download and run the latest Alchemy Plugin installer. Check to see if the presets and samples that are available have a higher version number than the ones that you originally downloaded.

To install new versions of the presets and samples, follow the instructions above for installing add-on sound banks.

Using the interface

While each group of functions in Alchemy is organized into a control panel or sub-panel described elsewhere in this Manual — see the table of contents in the right-hand margin — several types of controls are found throughout the interface. These include **knobs**, **buttons**, and **value fields**.

Alchemy also provides more specialized types of controls, such as the MSEG breakpoint editors and the graphical Source Editors. Each of these is explained separately on its own page in the Manual (MSEG, Source Edit, and so on).

Working with knobs

Many parameters are controlled via **knobs**. When you adjust a knob, its value is displayed and updated in the large Parameter Value display towards the top of the interface. Several useful techniques let you work with greater speed and accuracy.

- To **inspect a value** (without changing it), simply **hover over a knob** (without clicking it).
- For **fine control** of a parameter, **shift-drag** its knob.
- For **coarse but rapid control** of a parameter, hover over its knob and roll your scroll wheel.
- To **reset any knob** immediately to its 12 o'clock position, **double-click** it.
- By default, all knobs in Alchemy respond to **linear dragging**: drag a knob upwards to increase it and downwards to decrease it.
    - You can **alt-drag** (PC) **or option-drag** (Mac) a knob to have it respond momentarily to **circular dragging**: drag a circular path around the knob, clockwise to increase it and counter-clockwise to decrease it. One nice application of this technique: you can set a knob **instantly** to the position you want by **alt- or option-clicking** that position.
    - If you prefer circular dragging as the default behavior, you can edit the AlchemyConfig.txt file. Insert the following line for circular default behavior: 'KnobCircular= 1'. To restore linear default behavior, change this line to 'KnobCircular= 0' — or delete the line entirely. AlchemyConfig.txt is located in the same folder as the Alchemy.dll file, within your VST Plugins folder (PC); or at /Library/Application Support/Camel Audio/Alchemy (Mac).
- To access a **contextual menu** offering various knob actions, **right-click or control-click** a knob.

Working with buttons and value fields

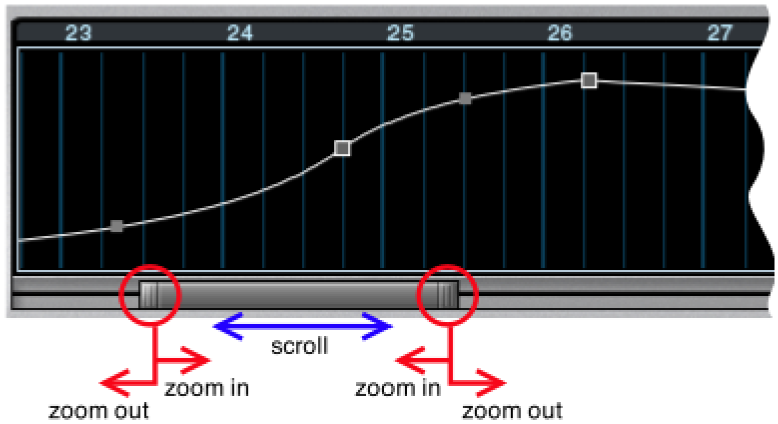
Operating **buttons** on the Alchemy interface is straightforward: simply click them. Many buttons behave as **toggles**: click a button once to turn a function on (button is lit) and click it again to turn the same function off (button is unlit). Some groups of buttons are **mutually exclusive**: turning one on turns another off. Some buttons **contain pop-up menus**: click one of these buttons and then choose a more specific command from the menu.

Value fields display their current value in a text field. Clicking in this field opens a **pop-up menu** from which you can choose a new value. Many value fields also provide **forward and back buttons**, so you can browse through the available choices.

Scrolling and zooming

Several of Alchemy's displays can be scrolled left and right to bring more controls or data into view, and many of these displays can also be zoomed in and out, so you can focus on small details or see more data at once.

- To **scroll**, click the scroll bar and drag it left or right.
- To **zoom in**, click a zoom handle at either end of the scoll bar and drag it inward. To **zoom out**, drag it outward.
- To **zoom all the way out**, exposing a maximum amount of data, double-click the scroll bar.



MIDI Learn

1. Right-click (control-click) any of Alchemy's knobs and choose 'MIDI Learn' from the contextual menu.

2. Move a control on your MIDI keyboard/controller.

Alchemy will automatically 'learn' the MIDI control and assign it to the knob. Now you can adjust the onscreen knob by operating the hardware control.

- Every one of Alchemy's knobs can 'learn' a MIDI control in this way.
- Alchemy's XY Pads and Remix Pad (see the Performance controls page) 'learn' in almost the same way, except that there are two options in the pop-up menu. 'MIDI Learn X' assigns a control to the pad's horizontal axis. 'MIDI Learn Y' assigns a control to the pad's vertical axis.
- Once a MIDI control is learned, the assignment persists each time you load Alchemy. You can undo an assignment by right-clicking the knob and choosing 'MIDI Unlearn' from the contextual menu.
- The current set of MIDI-learn assignments is stored in a file called 'MidiMap.txt', which is located in the same folder as the Alchemy.dll file, within your VST Plugins folder (PC); or at
/Library/Application Support/Camel Audio/Alchemy (Mac).

Title bar

Alchemy's title bar provides tools for preset management and a few additional functions.

Bank, Category, and Preset Fields

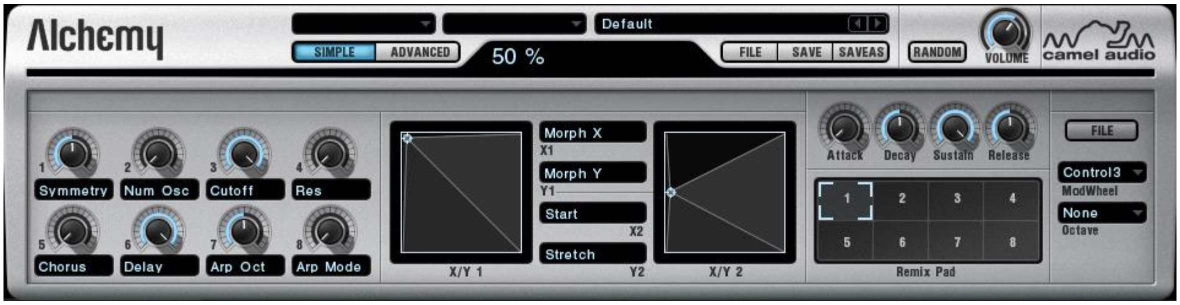
At the top and in the center are three fields, from left to right: Bank, Category, and Preset. When Alchemy is first installed, the available banks are Factory and User. Categories may vary from one bank to another, but the standard choices found in the Factory and User banks include 'Arpeggiated', 'Bass', 'Brass', and so on. Of course the list of presets is unique in each category. You can browse through the available presets quickly using the forward and back arrows adjacent to the Preset field. Note that the categories in a given bank, and the presets in a given category, are listed alphabetically.The organization of Banks and Categories mirrors the file structure on your computer. A category is a folder containing .acp preset files (plus associated audio data), and a Bank is a folder containing categories. So you can create your own categories and banks just by creating additional folders in the proper locations. On a PC, Bank folders go in \Alchemy\Presets. On a Mac, Bank folders go in /Library/Application Support/Camel Audio/Alchemy/Presets. (On both PC and Mac, new Bank folders sit alongside the existing Factory bank.)

Parameter value display

Beneath the Bank/Category/Preset fields is a large numerical display that shows the current value of the active or selected control, calibrated in appropriate units (e.g. '50%', '5 semis', '2859 Hz', and so on).Alchemy also uses this display for occasional messages (e.g. to confirm that a Save command was executed successfully) and to display a sound-designer credit when you first load each preset.

Simple and Advanced buttons

When the **Simple** button is active, a compact interface is displayed, consisting of the Title bar and the Perform section (see below). When the **Advanced** button is active, a larger interface is displayed in which all sections of Alchemy are accessible. (See the Overview page for a quick tour of the entire interface.)



File, Save and Saveas buttons

The **FILE** button opens a pop-up menu containing several commands useful for preset management:

- **Load** — opens a dialogue box where you can select a preset file (*.acp) to load.
- **Save Consolidated** — saves a copy of the active preset along with copies of any WAV files it depends on.
- **Clear** — Initializes Alchemy, giving you a basic starting point for creating your own presets. (The initialised preset is a basic sawtooth wave.)

The FILE button pop-up menu contains additional commands that you can use whenever your computer is connected to the internet:

- Check for Update — checks if you are using the latest available version of Alchemy.
- Download Free Presets — visits the Camel Audio Downloads webpage, where registered users have access to additional banks of presets.
- Watch Tutorial Videos — visits the tutorial video section of the Camel Audio website.
- Read Wiki Manual — points your web browser to Alchemy's online User Manual. The advantage of the web-based manual over the PDF version distributed with Alchemy is that it can revised as often as necessary to incorporate corrections and descriptions of new features added in each update.
- Support Forum — points your web browser to the Camel Audio support forum (http://www.kvraudio.com/forum/viewforum.php?f=32), where you can ask questions and post comments about Alchemy and other Camel Audio products.
- Camel Audio Shop — points your web browser to a page where you can read about, and purchase, other Camel Audio products.

The **SAVE** button saves your changes to the currently active preset.The **SAVEAS** button opens a dialogue box where you can choose a file name (*.acp) and a location to save the currently active preset.

> ***Note: If a preset has associated additive and/or spectral data, that data will be stored in a separate file (*.aaz). The first part of the filename is the same as the name of the saved preset, which ensures that the preset and the associated additive/spectral data appear side-by-side in an alphabetical directory. It is recommended that you avoid renaming .aaz files externally (i.e. from your operating system); Alchemy may fail to locate and load these files unless their original names are preserved. While renaming a whole preset externally is less risky, it may lead to duplication of .aaz files to reflect the new name. In general, it's best to rename presets by using Alchemy's SAVEAS function. Once a preset is successfully saved under its new name, you can safely discard the old preset (*.acp) file.***

Random button

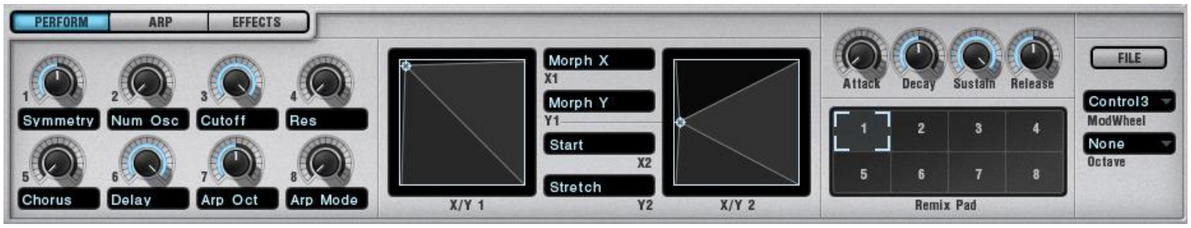
Creates a random preset according to the currently selected preset category.

Volume knob

Boosts or cuts the overall preset volume. This adjustment is made at the very end of the signal path, so it controls the effects as well as the dry portion of the signal, and it acts on all voices at once. (Therefore, it's less common to modulate the Volume knob compared to the Amp knob on the Master page.)

Performance controls

When the **PERFORM** button is illuminated, the Performance controls sub-page is displayed in the Perform / Arp / Effects section.



The Performance controls sub-page provides eight knobs, two XY pads, and a set of 'ADSR' envelope knobs - a total of 16 controls. These can be assigned to modulate parameters in the same way as Alchemy's other modulation sources.

The eight knobs can each be named, by clicking in the field below a knob and typing a name. Similarly, a pair of fields is provided for naming the X and Y axes of each XY pad.

Right-clicking (control-clicking) any perform knob opens a pop-up menu with the usual MIDI Learn/Unlearn commands, plus the following more specialized functions:

- **Delete Modulation** — offers a list of all the targets receiving modulation from the knob; selecting an item in this list deletes the corresponding modulation assignment.
- **Swap With** — allows you to exchange all the modulation assignments between the knob you've clicked and any other performance control selected from a list.
- **Copy Setting to all Snapshots** — updates all eight snapshots so that the current value of the knob you've clicked replaces the stored value of that knob in each snapshot. For instance, if you want the set the master Sustain knob to 100% in every snapshot, simply turn the knob to the 100% position, right-click it, and choose this command to apply the new setting to all eight snapshots.

Right-clicking (control-clicking) an XY pad opens a pop-up menu containing similar commands for both the X and the Y axis.

Remix Pad

An Alchemy preset contains eight 'snapshots' of Performance control settings, numbered 1 to 8, and accessible via the numbered grid at the far right of the Performance controls sub-page. By default, snapshot 1 is selected, and outlined with a white box.

To make a new snapshot, set each of the Performance controls as desired, click the FILE button, choose ‘Store Snapshot’, and select a number from 1 to 8. (Or choose ‘All’ to store the current Performance control settings in all eight snapshots. Note that this will overwrite all eight of the existing snapshots.)

You can ‘morph’ between Performance snapshots in real-time by clicking and dragging the white box around the Remix Pad. The Performance control knobs will update smoothly as you go.

> ***Note: to select a snapshot precisely, double-click it. (See also ‘Keyswitching’, below.)***

Auto Assign and other controls

The **FILE** button on the Performance sub-page opens a pop-up menu with the following options:

- ‘Copy Snapshot’ stores all the Performance settings of the currently selected snapshot in a buffer.
- Paste Snapshot’ pastes all the Performance settings from the buffer to the currently selected snapshot.
- ‘Copy 1 to All’ copies all the Performance settings of snapshot 1 into snapshots 2 to 8.
- ‘Clear’ cancels all modulations by the Performance controls, resets all the controls to zero, and clears the knob and pad name fields.
- ‘Auto Assign All’ automatically assigns all the Performance controls to appropriate parameters in the current preset. It also intelligently creates a random set of snapshots, as variations on the settings in snapshot 1. This is the quickest way to get started with the Performance controls. Try it!
- ‘Auto Assign Empty’ automatically assigns all the unused Performance controls to appropriate parameters in the current preset. It doesn’t interfere with any Performance controls assignments you have already made, and it doesn’t create snapshot settings.
- ‘Store Snapshot’ lets you store the current Performance control settings in your choice of the snapshots 1 to 8.

Keyswitching

It's also possible to switch between Performance snapshots using your MIDI controller keyboard.

From the **Octave** pop-up menu, select a MIDI note name (e.g. 'C4'). The first eight notes of that octave (including both white and black keys, e.g. C4–G4) will be assigned to snapshots 1–8. Hit one of the assigned keys on your controller to select the corresponding snapshot.

> ***Note that MIDI notes assigned to snapshots in this fashion will not trigger normal notes in Alchemy. To ensure that notes play normally across the entire range of MIDI note numbers, choose 'None' from the Octave menu.***

Modwheel

The Modwheel pop-up menu links the modwheel to one Performance control of your choice (any knob, or the X or Y axis of either pad).

Therefore assigning the modwheel to modulate any target in Alchemy is a two step process:

1. Assign one of the Performance controls to modulate the target.
2. Link the modwheel to the Performance control you assigned in step 1, by selecting that control in the Modwheel pop-up menu.

Recommended performance control assignments

The performance controls can, of course, be assigned any way you find useful. But if you are a sound designer creating presets for others to use, it is recommended that you follow the same guidelines that were established for Alchemy's factory presets. Here is a summary of those guidelines:

- Make sure that all perform controls are assigned to something and have suitable names. Control names should be no more than 10 characters long, and each word should start with an upper-case letter.
- The 'Auto Assign' feature (accessed via the Perform section's FILE button) often gives the best starting point; but creativity — including the assignment of knobs to multiple parameters and the use of ModMaps for custom scaling — is encouraged.
- There is a **standard Perform layout**, and sound designers are urged to follow it. Occasionally, a preset design is driven by some concept that requires special

performance controls. (For instance, if you create a preset that distributes four different drum sounds across the keyboard, it might be important to designate two knobs per drum sound for tone controls.) But in the vast majority of cases, the standard layout should be followed as closely as possible:

  - **Knobs 1–2 are for timbre, pitch, other Source parameters.** There is a degree of flexibility with these two controls, but if there are assignments for Source parameters, this is the primary place they should be.
  - **Knobs 3–4 are for filter cutoff and resonance.** This pair of controls should be the primary place users go to tweak filtering. If your preset design doesn't involve a filter, considering adding one in its fully open state in the Effects stage; then users get your intended design plus the option to apply filtering.
  - **Knobs 5–6 are for effects levels or other effects parameters.** It's always good to allow users to tailor the effects levels to suit their needs.
  - **Knobs 7–8 are for controlling a preset's rhythm or movement.** There is a degree of flexibility with these two controls, but they are often assigned to LFO or Arpeggiator parameters, or to anything that causes or influences movement in the sound. Alternatively, **Knob 8 may be assigned to Master Amp or Master Volume** in order to allow the level of each Remix snapshot to be kept in balance.
  - **XY square 1 is for morphing among Sources.** If more than one Source is used, this XY control is the place to assign control of morphing or crossfading. (If not, the assignment of this square is flexible.)
  - **XY square 2 is flexible.** It is often used for control over additional effects parameters, or for features that are unique to a particular preset.
  - **The Attack/Decay/Sustain/Release knobs are master envelope controls.** They should normally control the full range of the corresponding parameters in AHDSR 1. (The 'Auto Assign Empty' feature can take care of this for you.) If you are using an MSEG rather than an AHDSR in the role of master envelope, these controls may be left unassigned.

- The **ModWheel** should be linked to the most playable performance control. The Remix Pad key-switching **Octave** should be set to 'Off', so that this feature doesn't interfere with normal playing across the entire keyboard. And on the Arp page, the arpeggiator's **Latch** mode should be set to 'Off'.

Arp

Alchemy offers a flexible and powerful **Arpeggiator**. To access it, click the **ARP** button in the Perform/Arp/Effects section towards the bottom of the interface. A large number of controls makes the Arpeggiator unusually versatile; but if you just want to create classic arpeggiator patterns, you'll find this very easy to do. (In fact, you can probably skip straight to the Examples at the bottom of this page!)

The Arpeggiator sub-page consists of three main sets of controls. On the left is a series of nine knobs that control the basic behavior of the Arpeggiator. In the middle is a set of buttons, pop-up menus, and knobs that let you manage one or more Sequencers. These are called 'Arp Sequencer' modules, because they are internal to the Arpeggiator and separate from Alchemy's normal Sequencers. On the right of the Arpeggiator subpage is a step editor for the Arp Sequencer modules. Below, you will find details about each of the basic sequencer controls, and an explanation of what the Arp Sequencer modules do. For more details about individual Arp Sequencer controls, see the explanation of the same controls on the page describing Alchemy's normal Sequencers.

Basic controls

The basic Arpeggiator controls consist of nine knobs. Here's what each of them does.

- **Mode.** Turns the Arpeggiator On and Off, and determines the order in which incoming notes are organized into a pattern. In addition to 'Off', you have the following choices:
  - 'Up' plays the current notes from lowest to highest.
  - 'Down' plays the current notes from highest to lowest.

- ‘Up/Down’ — lowest to highest and back again.
- ‘Down/Up’ — highest to lowest and back again.
- ‘As Played’ — plays the current notes in the order they were originally played.
- ‘Random’ — plays the current notes in a random, non-repeating order.
- ‘Chord’ — plays all of the current notes simultaneously as a chord. In this mode, the chords you play will ‘pulse’ according to the Arpeggiator’s rate and rhythm, as determined by the the Arp Sequencer settings.

Note: Like most knobs throughout the Alchemy interface, Mode is a mod target. In several of the factory presets, a Perform knob is set up to modulate Mode, so you can easily turn the Arpeggiator on and off during a performance.

- **Latch.** When Latch is Off, the current Arpeggiator pattern will stop playing when you lift your hands off the keyboard. When Latch set to ‘Hold’ or ‘Add’, the Arpeggiator ‘holds’ notes for you, so the current pattern will continue to play when you lift your hands off the keyboard. If you then play one or more new notes, two results are possible:
  - In ‘Hold’ mode, the new notes are organized into a new pattern, which replaces the existing pattern.
  - In ‘Add’ mode, the new notes are added to the existing pattern. (Try combining ‘As Played’ mode with the ‘Add’ latch setting to create an interactive step sequencer with up to 128 steps!)
- **Amp.** The Amp knob determines the velocity of notes played by the Arpeggiator (but see also the KeyVel knob). It can be modulated to create variations in note velocity as the Arpeggiator plays.

Note that Arp Sequencer 1 is assigned by default to modulate the Amp knob. Normally you should not remove this modulation assignment. Even if you don’t need the step values of Arp Sequencer 1 to determine the velocities of notes played by the Arpeggiator, this modulator plays a more fundamental role: the rising edge of each Arp Sequencer step is the Arpeggiator’s cue to play the next note in its current pattern.

- **Tune.** The Tune knob applies a pitch offset to all of the notes played by the Arpeggiator. For example, if you set Tune to 12 semitones, the Arpeggiator will play patterns an octave higher than the MIDI input. Tune is also available as a target to be modulated (e.g. by an Arp Sequencer). This allows you to create a variable pitch offset as the Arpeggiator plays.

- **Pan** sets a position in the stereo field for all the notes played by the Arpeggiator. It can be modulated to create variable panning as the Arpeggiator plays.
- **Split.** The keyboard can be split so that notes below a certain point are fed to the Arpeggiator, while notes above that point play normally. Split determines the highest note to be included in the Arpeggiator pattern; set it to the maximum value ‘g8’ to arpeggiate all of the MIDI input.
- **KeyVel.** Arp Sequencer 1 modulates the Arpeggiator’s Amp knob by default, and when KeyVel is 0%, the step values of Arp Sequencer 1 fully determine the velocities of notes in the Arpeggiator pattern. When KeyVel is 100%, these velocities are determined instead by the incoming MIDI data, so if you strike a key hard the corresponding note in the Arpeggiator pattern will have a high velocity level. You can also set KeyVel to intermediate values to blend the Arp Sequencer step values with the velocities in the incoming MIDI data.
- **Octave** determines whether the Arpeggiator pattern is played only at its original pitch level, or is repeated across additional higher octaves.
- **Source.** When Source is set to ‘All’, the Arpeggiator pattern is played by all sources (assuming they are all turned ON). You can also restrict the Arpeggiator pattern to any one source by setting this knob to a value of 1–4; then the other three sources will play the incoming MIDI data normally.

Using the Arp Sequencer modules

Arp Sequencer modules work just like normal Sequencer modules in Alchemy. They have a more specialized job to do, however, because they are internal to the Arpeggiator and are designed to modulate its basic controls (which are described above).

> ***Notice that if you right-click a knob in most sections of Alchemy (e.g. a Cutoff knob in the Main Filter section) and peek at the ‘Add Modulation’ options in the contextual menu, your choices include ‘Sequencer’. But if you right-click one of the Arpeggiator’s basic controls (e.g. its Pan knob), your choices include ‘Arp Sequencer’ instead.***

By default, the Arpeggiator is configured with just one Arp Sequencer, and for most purposes this is all you will need. Arp Sequencer 1 modulates the Arpeggiator’s Amp parameter. This modulation assignment enables the Arpeggiator to ‘watch’ Arp Sequencer 1: whenever the Arpeggiator sees the rising edge of another Arp Sequencer step, it plays the next note of its own current pattern.

Various settings in the Arp Sequencer allow you to adjust the Arpeggiator’s behavior.

- **StepDur.** Choose a different ‘StepDur’ to make the Arpeggiator play faster or slower.
- **Shuffle** lets you create various ‘swing’ effects. Setting a Shuffle value greater than 0% increases the duration of the odd-numbered step (1, 3, 5, …) and decreases the length of the even-numbered steps correspondingly.

  Note: For classic ‘swinging sixteenth notes’, set StepDur to 1/4 and try a Shuffle value of around 20% (light swing) to 30% (heavy swing). You can get more extreme effects using even higher Shuffle settings; at 100%, the even-numbered steps get so short that they disappear entirely!

- **Sustain** sets the length of each note played by the Arpeggiator, as a percentage of the StepDur. Lower values of Sustain will give you more ‘staccato’ results. (You may not hear a staccato effect, however, if your preset has a long release time.)

The Arpeggiator’s basic controls and the Arp Sequencer settings described above add up to a well-rounded Arpeggiator — one that can create a variety of patterns using notes played on the keyboard or other MIDI input, with controls in the Arp Sequencer for speed (StepDur) and rhythmic feel (Shuffle, Sustain). If you want even more options, you’ll find them in the Arp Sequencer step editor.

Using the Arp Sequencer step editors

When an Arp Sequencer modulates the Arpeggiator Amp, its step editor can be put to a number of uses. (Recall that Arp Sequencer 1 modulates Amp as part of the default configuration of the Arpeggiator.)

- You can edit the Arp Sequencer step **values** to create a pattern that the Arpeggiator will apply to the velocities of the notes it plays.
- If you set the value of a step to 0%, the Arpeggiator **won’t play a note** at that step. (There’s no ‘rising edge’ for the Arpeggiator to see when the step value is 0%; therefore it doesn’t trigger a note.) This creates a gap in the rhythmic pattern — a ‘rest’, in musical terms.
- Similarly, you can ‘tie’ one step to the next by shift-clicking below it. A small ‘chain-link’ symbol appears below each tied step. When two or more steps are tied together they behave like one longer step; only the first step triggers a note in the Arpeggiator, while subsequent tied steps provide sustain.

- You can edit the Arp Sequencer step **lengths** to create a pattern of shorter and longer note lengths. These values are combined with the overall note length determined by the Arp Sequencer **Sustain** control.
- You can edit the Arp Sequencer **swing** at each step to create subtle (or not-so-subtle) variations in timing. Each swing value ranges from 0 to 2; the middle value of 1 represents normal timing, while smaller values play earlier and larger values play later than normal. These swing values are combined with the overall timing pattern determined by the Arp Sequencer **Shuffle** control.

You can apply additional sets of Arp Sequencer steps to other basic controls in the Arpeggiator, such as Tune and Pan. Simply right-click on one of these knobs and choose 'Add Modulation' > 'Arp Sequencer' > 'New Arp Sequencer'. (Indeed, that's an early step in the example entitled 'Creating a step sequencer', found at the bottom of this page.)

Note that 'New Arp Sequencer' creates a new set of Value, Length, and Swing patterns that you can edit to suit your wishes; but all of the Arp Sequencer modules share a single set of controls such as StepDur, Shuffle, Sustain, Trigger status, and NumSt (number of steps). Alchemy's normal Sequencer modules work differently: each one has an independent set of controls such as StepDur.

Importing from a MIDI file

Alchemy's MIDI-import capabilities are described on the Sequencer page. The same **Import Velocity**, **Import Note**, and **Import Groove** commands are available here in the Arp Sequencer modules. An additional command, **Import All** is also provided; it sets step values in Arp Sequencer 1 based on extracted velocity data, sets swing values in Arp Sequencer 1 based on extracted groove data, sets step levels in Arp Sequencer 2 based on extracted note data, routes Arp Sequencer 1 to the Arpeggiator's Amp knob with a depth of 100%, and routes Arp Sequencer 2 to the Arpeggiator's Tune knob with a depth of 24 semis. The resulting configuration allows the Arpeggiator to play back the imported MIDI notes, velocities, and groove timing faithfully.

Example: Creating classic arpeggiator patterns

- Initialize Alchemy by clicking **FILE** in the Title bar and choosing 'Clear' from the pop-up menu.
- Switch the Perform/Arp/Effects section to **Arp** at the bottom of the interface, and play and hold a chord. Set **Mode** to 'Up/Down' and **Latch** to 'Hold'. Now when you play a chord, it will be processed by the Arpeggiator, which continues to play when you release the chord.
- Next, as the Arpeggiator continues to play, you can explore additional Arpeggiator parameters:
    - Try changing the **Mode** to a value other than 'Up/Down' — 'Up', 'Down', 'Down/Up', and 'Chord' are among the most popular additional Arpeggiator modes.
    - Try setting **Octave** to a value greater than one in order to have the Arpeggiator pattern play in more than one octave range.
    - Try changing the Arp Sequencer **StepDur** from its default value of '1/4' to '1/2' (so a note plays every half beat), to '1/8' (eight notes per beat), and to other values. Restore a value of '1/4' or greater before trying the next step.
    - Experiment with the Arp Sequencer **Shuffle** and **Sustain** parameters.
- While the Arpeggiator continues to run, you can also adjust settings in other sections of the interface. For instance, try the following adjustments:
    - Source A **VA waveform** = 'Basic' > 'Square'; **NOsc** = 2.
    - **AHDSR 1**: ***Decay*** = 0.50 sec; ***Sustain*** = 0%; ***Release*** 1.5 sec.
    - **Main Filter 1**: ***Cutoff*** = 550 Hz; ***modulate*** Cutoff with AHDSR 1 (depth = 30%).

Example: Creating a step sequencer

- Initialize Alchemy by clicking **FILE** in the Title bar and choosing 'Clear' from the pop-up menu.
- Switch the Perform/Arp/Effects section to **Arp** at the bottom of the interface, and set **Mode** to 'Up' and Latch to 'Hold'. Now play a single note and the Arpeggiator will play that same note repeatedly as you work through the rest of this example. (If this drives you crazy, set Latch to 'Off' instead!)

- Right-click on the Arpeggiator's **Tune** knob and choose 'Add Modulation' > 'Arp Sequencer' > 'New Arp Sequencer'.
- The Tune knob's mod rack appears in Alchemy's MOD section, and 'Arp Sequencer 2' can be seen in the first slot of the rack.
- Adjust the modulation Depth knob, to the right of this slot, to a value of 24 semitones in order to program a step pattern spanning two octaves.
- Returning to the Arpeggiator sub-page, set the 'Snap' value for the Arp Sequencer step editor to 1/24 in order to snap step values to semitones (since the modulation depth is 24 semitones).
- Ensure that the step editor view is set to Value (rather than Length or Swing, which are only useful when you're modulating Amp). Now you can create a pattern of pitches by editing each step value as you desire.

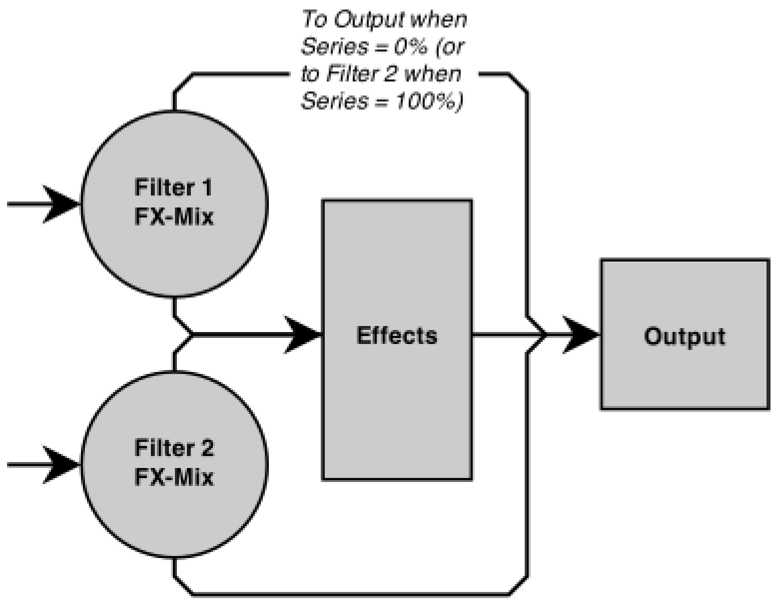
Make it stop!!!

If the Arpeggiator keeps playing and you want it to stop, do one of the following things:

- Set the Arpeggiator **Mode** knob to 'OFF'.
- Set the Arpeggiator **Latch** knob to 'OFF'.
- **Stop** (or Play and then Stop) your host's transport.

Effects

Alchemy's Effects module provides a powerful multi-effects processor, offering up to five high-quality effects simultaneously. The available effects include all of the choices found in Camel Audio's popular CamelPhat and CamelSpace multi-effects plug-ins as well as its Cameleon 5000 synthesizer. Several new effects, including a highly realistic Acoustic Reverb, round out the choices.



The Effects module receives its input from the two Main filter modules, in varying amounts, according to their respective FX-Mix settings.

It sends its output directly to Alchemy's main output.

Note: the Effects module's output is mixed together with the 'dry' output from the Main filter modules before arriving at Alchemy's main output.

Global page controls

When the **EFFECTS** button is illuminated, the Effects sub-page is displayed in the Perform / Arp / Effects section.

The left-hand side of the Effects sub-page contains the **effects rack**, with five slots into which effects can be loaded. The signal flows through these loaded effects **from the top to the bottom** of the rack, so you can get different results by loading the same effects in different orders.

Clicking the **FILE** button above the effects rack gives you access to a pop-up menu with several useful commands.

- The **Load, Save**, **Copy**, and **Paste** commands act on the entire contents of the effects rack and on the settings of each loaded effect.
- The **Clear** command removes all the loaded effect effects, giving you an empty effects rack.
- The **Randomize** command randomizes the settings of all the currently loaded effects.

Each slot of the effects rack consists of an **on/off button** and a **selection field**.

- To **bypass an individual effect**, click its on/off button (lit = on, unlit = off).
- To **remove an effect** from the rack, click its selection field and choose 'None' from the pop-up menu.
- To **load an effect** in the rack, replacing any previously loaded effect in the same slot, click the selection field and choose from the pop-up menu.

  Each effect can be used in only one slot, but a second instance of several effects is provided. (For instance, if you've already used the Delay1 effect, it will be greyed out in the pop-up list of effects available in the other effects rack slots; but you can still choose Delay2 further down in the list.)

- To **change the order of the loaded effects**, right-click (control-click) a selection field and use the 'Move Up' and 'Move Down' commands.

Controlling individual effects

For each effect loaded in the effects rack, there is a corresponding **control panel** in the scrolling window on the right-hand side of the Effects sub-page. (The top-to-bottom order of the rack is matched by the left-to-right order of the window.)

Many of the effects control panels include **FILE** buttons. Clicking one of these buttons gives you access to a pop-up menu with **Load**, **Save**, **Copy**, **Paste**, and **Clear** commands, which apply to the settings of the individual effect. (Note that the ‘Clear’ command restores an effect’s default settings; it doesn’t bypass the effect or remove it from the effects rack.)

Acoustic Reverb. High quality, full-featured reverb effect.

- **Time** — controls the length of the reverb tail; values up to 20 sec are possible.
- **PreDelay** — sets amount of initial delay before diffuse reflections begin.
- **Size** — determines the dimensions of the simulated space.
- **Width** — determines how much reflections are spread across the stereo field.
- **Diffusion —** determines the density of the early reflections.
- **Gate** — sets a threshold below which the reverb tail will be gated. **Attack** shapes the beginning of the gated tail, while **Decay** shapes its end.
- A built-in ***EQ*** lets you modify the frequency content of the wet signal. **LoFreq** sets the frequency of a low shelf that you can boost or cut with **LoGain**. **HiFreq** and **HiGain**, similarly, work like a high shelving EQ.
- There are four controls for ***damping*** (frequency-specific losses). **DampLoF** sets the frequency of a low shelf that you can damp by increasing **DampLoAmt**. **DampHiF** and **DampHiAmt**, similarly, provide high-shelf damping.
- **Variation —** provides subtly different ‘colors’ of reverberation.
- **Quality** — at lower settings of Quality, the Acoustic Reverb uses less CPU. At higher settings, it uses more CPU in order to provide a denser, smoother tail. When you need to conserve CPU, don’t be afraid to try lower settings of Quality; for many types of material these settings still sound excellent.
- **Mix —** determines the wet/dry balance (0% = dry only; 50% = equal mix; 100% = wet only).

Camel Reverb. Reverb effect featuring the CPU-efficient algorithm used in Camel Audio’s popular CamelSpace effect and Cameleon 5000 synthesizer.

- **PreDelay** — sets amount of initial delay before diffuse reflections begin.
- **Damping** — higher values mean more high-frequency loss in the reflections.
- **HighCut** and **LowCut —** set the frequencies above and below which the wet signal is cut.

Take care not to inadvertently cut the entire wet signal by setting LowCut above HighCut.

- **Size** — determines the dimensions of the simulated space; larger values mean longer reverb times.
- **Mix** — determines the wet/dry balance (0% = dry only; 50% = equal mix; 100% = wet only).

Delay1 / **2**. Stereo delay with dual-filtered feedback.

- **L/R Rate** — sets the delay time, ***in msec*** when the **SYNC** button is off or ***in beats*** when the **SYNC** button is on.
- **L/R Offset** — adds a small additional amount of delay, so you can adjust the 'feel' of tempo-synced delays.
- **L/R Feedback** — determines how much of the delayed signal is fed back to the input of the delay.
- **Filter A, Filter B** — filters the delayed signal without affecting the dry signal. Toggle one or both filters with the **A** and **B** buttons (button illuminates when filter is on). Select a ***type*** for each filter via the selection fields. Each filter has **Cutoff** and **Resonance** controls.

  Note that Filters A and B are configured in series, and the left and right channels pass through both filters. Also please be aware that some filter types and settings — such as medium-to-high Resonance — boost the delayed signal, so you should reduce the amount of Feedback in order to compensate for the boost.

- **SYNC** button — synchronizes delay rates with the host tempo (See **L/R Rate**, above).
- **MONO** button — mixes the left and right input channels down to mono and feeds the result to both the left and the right channels of the delay. The 'dry' portion of the signal remains in stereo.
- **Crossover** — determines stereo placement of the feedback signal. (At 0%, left feeds left and right feeds right; at 50%, each channel is fed to both inputs; at 100%, left feeds right and right feeds left.)
- **Initial Pan** — determines the stereo placement of the initial delayed signal (prior to Feedback). Typically, you would set Initial Pan to 0% or 100% when Crossover = 100%, and leave Initial Pan centered 50% otherwise.
- **Mix** — determines the Wet/Dry balance (0% = dry only; 50% = equal mix; 100% = wet only).

  See the example at the bottom of this page for an explanation of how to use Crossover and Initial Pan for a classic 'ping-pong' delay effect.

Mod FX1 / 2. Short delay with built-in LFO modulation, useful for **chorus**, **flanging**, and related effects.

- **Delay** — sets the base delay time. The shortest values are useful for flanging, values in the range of 10 to 40 msec are useful for chorus, and longer delay times can produce a variety of metallic and buzzing effects.
- A built in LFO drives modulation of the base delay time, and controls are provided for the LFO **Rate** and the **Depth** of modulation. Faster Rates and smaller Depths are characteristic of chorus effects, while slower Rates and greater Depths are typical for flanging.
- **Feedback** — mixes the delayed signal back into the input. Medium to medium-high settings are common for flanging effects, while chorus tends to use little or no feedback.
- **Stereo** — spreads the delayed signal across the stereo field.
- **Mix** — determines the wet/dry balance (0% = dry only; 50% = equal mix; 100% = wet only).

Distortion1 / 2. Distortion effect with multiple algorithms that can be used simultaneously.

- **Tube** — simulates the warm distortion effect of an overdriven tube amp.
- **Mech** — produces a ‘nastier’, more intense flavor of distortion
- **Bit Crush** — a lo-fi digital flavor of distortion.
- **Xcite** — refreshes the high-frequency range.
- **Post Gain** — some distortion types, especially Tube and Mech, can boost the signal significantly; this control allows you to trim the output level to compensate.

Three Band EQ1 / 2. Offers three bands of parametric EQ. Each band has an identical set of controls.

- **LoGain** (**MidGain**, **HiGain**) — *s*ets the amount of boost or cut to the Lo (Mid, Hi) band.
- **LoFreq** (**MidFreq**, **HiFreq**) — sets the center frequency of the Lo (Mid, Hi) band.

  Note that LoFreq, MidFreq, and HiFreq can each be set to values from 16 Hz through 16744 Hz, so it isn’t necessary (although it may be useful) to tune Mid above Lo, or Hi above Mid…

- **LoBW** (**MidBW**, **HiBW**) — sets the bandwidth of the Lo (Mid, Hi) band.

Bandpass Filter1 / 2. Adjustable-width bandpass filter; the rejected portion of the signal can be recovered at a later point in the signal path using the Band Reject (or Band Reject 2) module.

- **Low**, **High** — set the lower and upper edges of the passband, respectively.
- **LowRes**, **HighRes** — add resonant emphasis to the passband edges.

Band Reject1 / 2. Only available when the Bandpass Filter1 / 2 is inserted upstream in the signal path. Allows you to mix in the portion of the signal that was rejected by the bandpass filter. A classic application is **multiband distortion**; see the example at the bottom of this page.

- **Mix** — determines the how much of the band-rejected signal is mixed back into the signal path, bypassing any effects that sit in between the Bandpass Filter and the Band Reject mix point.

MM Filter1 / 2. Multi-mode filter, offering all the same filter types as Alchemy's Main Filters. Note that each voice in Alchemy is processed by its own Main Filters, while all the voices are mixed together before passing through the MM Filter and other effects.

- The **filter types** and the **Cutoff**, **Resonance**, and **Drive** controls are the same as for Alchemy's Main Filters.
- **Mix** — determines the wet/dry balance (0% = dry only; 50% = equal mix; 100% = wet only). Typically this is left at 100%.

Bass Enhancer. Easy-to-use, bass-boosting EQ, recommended for use with bass and low-frequency percussion sounds.

- **Amount** — controls the intensity of the bass enhancement.
- **Tune** — adjusts the frequency band to which the boost is applied.
- **P** (phat mode) button — adds a saturation effect, for a different 'flavor' bass enhancement. For more neutral/transparent results, leave phat mode off.

Compressor. Easy-to-use dynamics processor based on a soft-limiting compression algorithm.

- **Amount** — controls the amount of compression. Because make-up gain is built in, larger amounts of compression result in greater apparent 'loudness'.
- **Release** — controls how quickly the compression effect subsides once the input signal falls below the threshold for compression.
- **P** (phat mode) button — adds a saturation effect, for a different 'flavor' of compression. For more neutral/transparent compression, leave phat mode off.

Gain. Allows for a variety of auto-gate effects when modulated by an LFO or sequencer.

- **Gain** — a setting of 100% preserves the input signal level, while lower values reduce this level, all the way to silence at 0%.

Pan. Allows for a variety of auto-pan effects when modulated by an LFO or sequencer.

- **Pan** — a setting of 50% preserves the balance between the left and right input channels. Lower values boost the left while cutting the right; higher values boost the right while cutting the left.

Example: Ping Pong Delay and Flanger

- **Initialize Alchemy** by choosing the 'Clear' command in the Title bar's FILE menu.
- Click the EFFECTS button to bring the Effects sub-page into view. Play a view notes at this point, just to get familiar with the **'dry' sound** of the default preset before any effects are applied.
- Click in the first (top) slot of the effects rack and choose 'Delay'. Then play a few more notes and notice that the Delay module echos what you play at a time interval of one beat (as determined by the host tempo). Now **bypass** the Delay by clicking the button at the left of the first slot in the effects rack.
- Next, click in the second slot of the effects rack and choose 'ModFX'. Then play a few more notes and notice that the default settings of the ModFX unit produce a pleasant chorus sound. Let's adjust the ModFX settings to achieve a flanger effect. Try the following adjustments:
    - Reduce the ModFX **Delay** time to approximately 0.0006 sec.
    - Set the modulation **Rate** to around 0.2 Hz, and set the modulation **Depth** to 0.003 sec.
    - Increase **Feedback** to 60% (or more, to taste).
- Now that the flanger effect is working for us, use the buttons at the far left of the effects rack to bypass the ModFX module and re-activate the Delay so we can work on setting up a ping-pong delay effect. Try the following adjustments to settings in the Delay module in order to pan delays alternately left and right:
    - Increase the left and right **Rate** settings from 1 beat to 1/2 beat.
    - Increase **Crossover** to 100%.
    - Reduce **InitialPan** to 0%.
- Now use the button at the far left of the effects rack to re-activate the ModFX module, so you can hear the flanger effect **combined** with the ping-pong delay. Notice that, on one hand, the ping-pong delay is most noticeable when you play short notes, but on the other hand, the sweep of the flanger is not very dramatic on these short notes. Let's

reverse the order of the two effects to see if the situation improves. **To swap the order of the two effects-rack slots**, right-click (control-click) the first slot and choose 'Move Down' from the contextual menu — or right-click (control-click) the second slot and choose 'Move Up'. In this new configuration, the sweep of the flanger can be heard across all the delayed copies of the sound, producing a more dramatic sweeping effect.

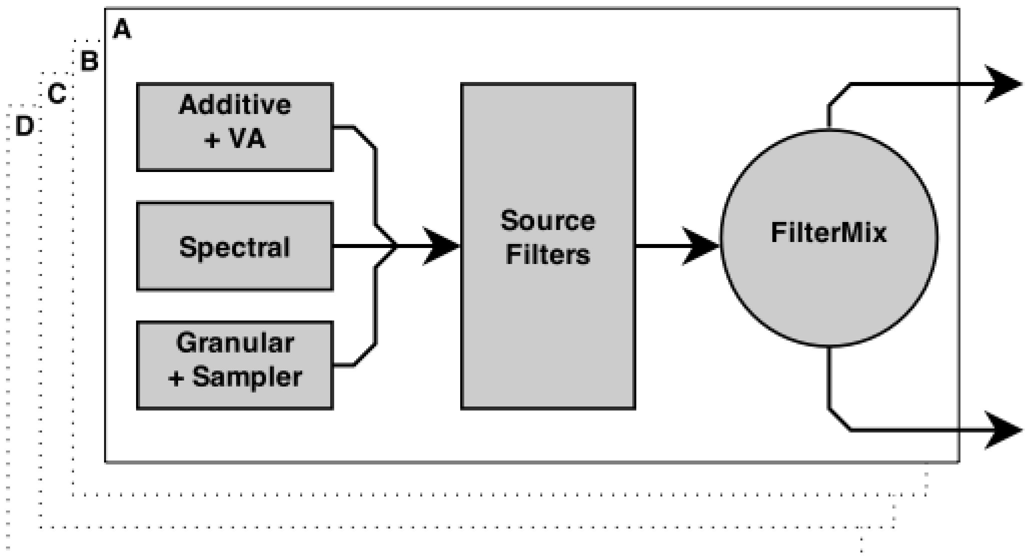
Example: Multiband Distortion using Bandpass Filter and Band Reject

- **Initialize Alchemy** by choosing the 'Clear' command in the Title bar's FILE menu. Then click in the Source A content field and choose 'Load Audio' > 'Factory' > 'Loops' > 'DrumLoops' > 'Silkloop'. Switch the Source A Granular element from Granular mode to Sampler mode for straightforward sample playback. (See 'Using Sampler mode' on the Granular page for details.)
- The 'Silkloop' drum pattern should loop smoothly when you play and hold middle C. Now **our goal** is to 'beef up' the kick drum with Tube distortion, while keeping the higher-frequency elements 'clean'.
- First, insert the **Bandpass Filter** into the first slot of the effects rack. While the drum pattern loops, reduce the High control until the kick drum is more-or-less isolated. (A value around 160 Hz works well.)
- Next, insert the **Distortion** module into the second slot of the effects rack. Turn the Tube control all the way up to 100%. (You can add an extra 'edge' to the resulting distorted kick sound by increasing the Bandpass Filter's HighRes knob — try a value around 20%.)
- Finally, it's time to bring the mid- and high-frequency portions of the drum back in via the **Band Reject** module. Simply insert this module in the third effects-rack slot, and turn the Mix control all the way up to 100%. As the drum loop continues to play, you can bypass and re-activate the Distortion module (using the button at the far left of its effects-rack slot) in order to confirm that this module is processing only the kick drum.

Source

Alchemy's sound-generating modules are called Sources.

There are four Sources available in an Alchemy preset (A, B, C, and D). Each Source is made up of an identical set of components:



The four Sources are entirely independent of one another, and can contribute to the overall sound in very different ways.

Elements

Each Source provides three sound-generating **elements** based on different methods of synthesis:

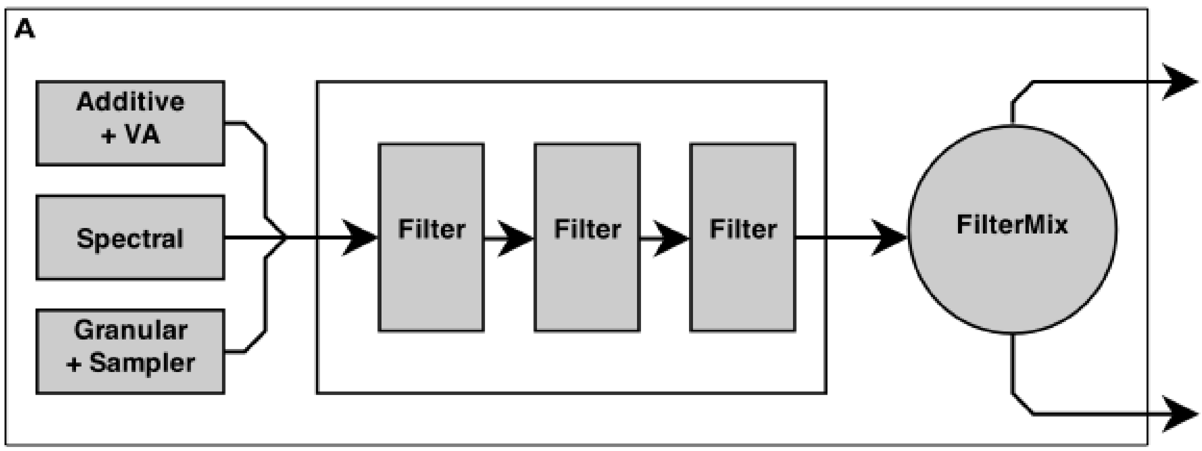
1. An Additive element, which can also be operated in virtual analog (VA) mode.
2. A Spectral element, which can also be used as a raw white and filtered noise source.
3. A Granular element, which can also be used for conventional sample playback (Sampler mode).

These can be toggled on or off independently, the only restriction being that when the Granular element is on, the Additive element can only be used in VA mode and the Spectral element can only be used as a noise source. To combine synthesis methods without facing this restriction, you can use multiple Sources.

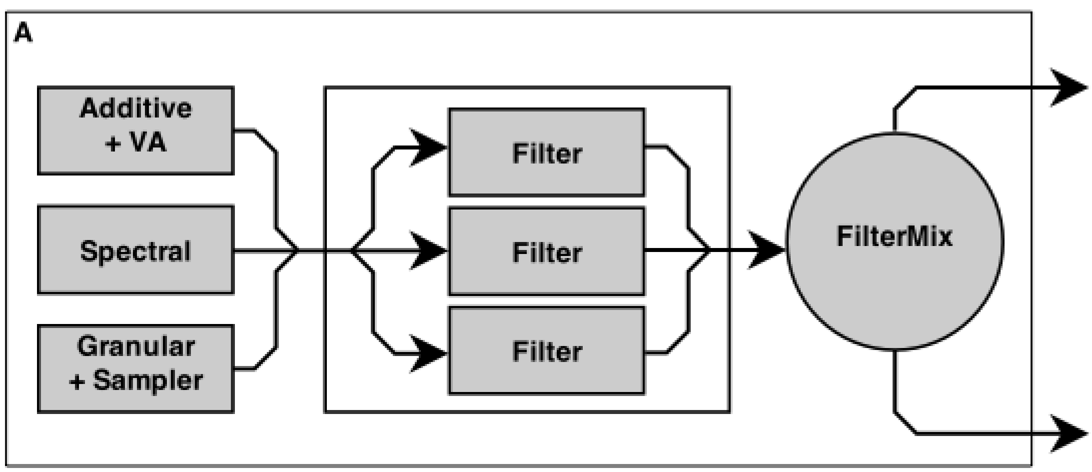
Source filters

The **Source filter** module provides three multi-mode filters, which can be configured either in **series** or in **parallel**. The Source filters allow you to filter each Source independently from the others. (Compare the Main filters, which process a mixture of all four Sources.)

With the Source filters in **series** the signal passes through the three filters one after the other, like this:



With the Source filters in **parallel** the signal is split and fed through the three filters simultaneously, like this:



> *Note: When the filter configuration is parallel and each filter is set to the 'Formant' type, a variety of Formant filter effects can be produced. See the example at the bottom of the* ***Filter*** *page for details.*

FilterMix

The **FilterMix** control works like a kind of two-channel mixer (more accurately, a crossfader).

- With the FilterMix control turned all the way to the left, all of the Source's output is fed to Main filter 1, and none of it to Main filter 2. (When you set the FilterMix knob this way, the parameter value display reads '100% F1 0% F2'.)
- With the FilterMix control turned all tthe way to the right, all of the Source's output is fed to Main filter 2, and none of it to Main filter 1. (When you set the FilterMix knob this way, the parameter value display reads '0% F1 100% F2'.)
- With the FilterMix control set half-way, equal amounts of signal are sent to Main filters 1 and 2. (When you set the FilterMix knob this way, the parameter value display reads '50% F1 50% F2'.)

Global page controls

At the top of the Global page's SOURCE section are five buttons: ALL, **A**, **B**, **C**, and **D**.

When the **ALL** button is illuminated, the Source master controls are displayed.

When the **A**, **B**, **C**, or **D** button is activated, the Source sub-page controls for the selected Source are displayed.

Source master controls

The Source master controls are a basic set of controls, duplicated for each of Alchemy's four Sources. These are a subset of the controls accessible from each Source sub-page.

- **On/off** button (labelled '**A**', '**B**', '**C**', '**D**'). Toggles the selected source on or off.
- **Source content** selection field. Displays the name of the current Source audio data. Clicking the field opens a pop-up menu with the following commands:
    - **Load Audio** — Quickly imports audio data (WAV, AIFF, or SFZ) into the Granular element, for playback in Granular or ordinary Sampler mode. Turns the Granular element on and the other elements off. If a SFZ file includes filter settings, configures a Source filter accordingly.
    - **Load VA** — Quickly loads waveform data into the Additive element, for playback in VA mode. Turns the Additive element on and the other elements off.
    - **Import Audio** — Opens a browser window for previewing sample data and importing it into the Granular, Additive, or Spectral elements. (See the Import page for details.)
    - **Load Source** and **Save Source** — Loads Source data from disk and writes it to disk. Uses special SRC file format. Loaded/saved data includes settings of all Source controls and a reference to loaded/imported audio data. (Source control ***modulations are not loaded/saved*** as part of the SRC file.)
    - **Copy Source, Paste Source** — Copies Source data to the clipboard and pastes it from the clipboard. Useful for duplicating the content and settings of one Source in another. Copied/pasted data includes settings of all Source controls and a reference to loaded/imported audio data. (Source control ***modulations are copied/pasted*** as part of the clipboard data.)
    - **Randomize Source** — Applies random variations to relevant Source controls, depending on which elements are active in the Source.
- **Amp** adjusts the level of the Source's output (–inf dB to 0 dB).
- **Tune** adjusts the level of the Source's output, in semitone increments.

  Note that a Fine Tune control is available on each Source sub-page.

- **Pan** positions the Source's output in the stereo field. (It works as a pan control if the source STEREO button is off, and as a left/right balance control if the source STEREO button is on.)
- **FiltMix** controls the routing of the Source output to main Filters 1 and/or 2, as described above.
- The **MORPH** controls (pop-up menu, **X** and **Y** knobs) are described on their own page.

Source sub-page controls

Each of Alchemy's four sources has an identical set of controls for more in-depth editing.

Several of the Source sub-page controls are duplicates of those found on the Source master panel. These include the **On/off** button, the **Source content** selection field, and controls for **Amp**, **Coarse Tune** (corresponds to 'Tune' on the master panel), **Pan**, and **FiltMix**. See details in the discussion of the Source master controls, above.

Each Source sub-page offers also offers several additional controls:

- The **Loop mode** selection field contains a pop-up menu with five choices:
  - **None** — Ignores the loop start and end points, and plays the entire sound once without looping.
  - **Continuous** — Plays from the beginning, enters the loop region, loops continuously in a forward direction while a note is played, and goes on looping during the release stage.
  - **Sustain** —Plays from the beginning, enters the loop region, loops continuously while a note is played, and exits the loop region to play the remainder of the sound during the release stage.
  - **Forward/Back** —Like Continuous, but plays the loop region alternately forward and backward. If jumping from the end of the loop region back to its beginning produces an unwanted discontinuity, Forward/Back mode may give you more continuous results.
  - **All** —Ignores the loop start and end points and loops the entire sound continuously.

Note. The loop start and loop end points can be edited in the Main view of the [Source edit](#) page. The Additive element in VA mode, and the Spectral element in WHITENOISE mode, use raw oscillators and noise sources rather than loopable data, so elements using these types of synthesis are not affected by the Loop mode setting.

- **Solo** button. When lit, isolates the Source by turning off all the other sources. Click again to restore the normal on/off status of other Sources.

  Note. If you save a preset with one Source in Solo mode, the result is a preset in which that one Source is on (but no longer in Solo mode) and the other Sources are off.

- **Edit** button. Gives you access to the Source edit page.
- The **KTRACK** selection field contains a pop-up menu with three choices:
  - **Key+PBend** — The pitch of the Source responds normally to MIDI Note data and PitchBend data.
  - **Key** — The pitch of the Source responds normally to MIDI Note data but does not respond to PitchBend data.
  - **Off** — The pitch of the Source does not respond to either MIDI Note data or PitchBend data.

  Note that the 'normal' response to PitchBend is set via the PitchBend Up and Down controls in Alchemy's Master section.

- **Stereo** button. Toggles stereo mode. If stereo mode is **off** and a stereo file is loaded/imported, only the left channel will be played. When stereo mode is **on**, loaded/imported sounds are played in stereo and various manipulations such as panning individual oscillators in the Additive element and panning individual grains in the Granular element are possible. (See the Additive and Granular pages for details.)
- **Fine Tune** adjusts the Source pitch in increments of one cent (one hundredth of a semitone).
- **Position**. Determines the position within the audio data from which playback starts. (0% = very start of data, 100% = very end.) See 'Modulating Position' below for more information.
- **Stretch**. In Additive, Spectral, or Granular mode, the Stretch setting determines the rate at which playback travels through the audio data. A setting of 100% represents the original playback rate. Higher settings (up to 500%) represent faster playback, while lower settings (down to 0%) represent slower playback.
  - Playback remains at the normal pitch regardless of the rate of travel.
  - Playback begins at Position and travels through the audio data on a path determined by the Loop mode. The Stretch setting determines the rate of this travel. (However, see 'Modulating Position' below for an alternative way to control the playback rate and path.)

- Setting Stretch to 0% 'freezes' playback at the position determined by Position.
- Stretch has no effect when the Granular element is set to ordinary Sampler mode.

Modulating Position

Position is a modulation target, which means you can control the path of playback through the audio data with a modulator.

- In the Granular element's ordinary Sampler mode, the modulation value at the Note-On moment determines an initial offset for the play position. Beginning at that position, the rest of the sound plays in a normal manner, but looped as if the Loop mode were set to 'All'.
- In Additive, Spectral, or Granular mode, Position can be continuously modulated.
    - Try setting Position and Stretch to 0% and modulating Position with a unipolar LFO, an MSEG, or a Perform control linked to the modwheel.
    - When Position is modulated and Stretch has a value greater than zero, the playback path is determined by a combination of the modulation value (whenever this value changes) and the 'normal' path (travelled at a rate determined by Stretch whenever the modulation value is static).

The example at the bottom of this page illustrates how to use modulation of the Position value to create a tempo-synced loop.

Elements and filters

A further four sub-pages are accessible from the bottom of each Source sub-page, which contain controls for the Source's elements and its Source filters.

For information on the element controls, see the Additive, Virtual Analog (VA), Spectral, and Granular pages.

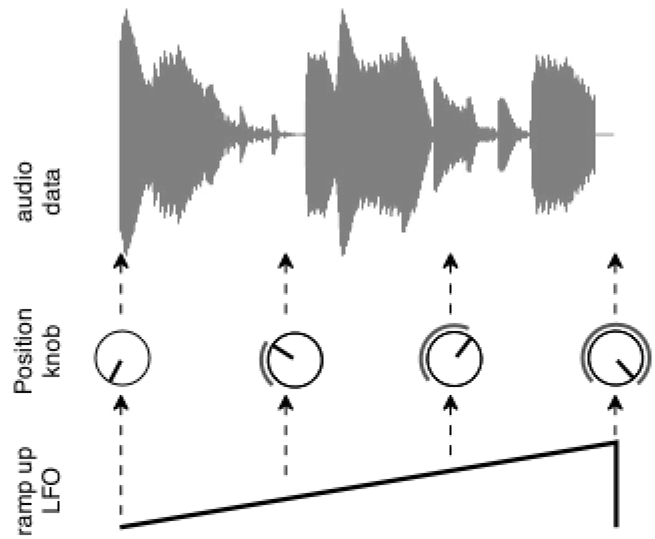
The Source filters are described above. For details on filter types and filter controls, see the Filter page.

Example: Tempo-synced loops

Synchronizing playback of looped audio with the host tempo is easy to achieve by modulating the Position parameter appropriately. This technique is possible with any synthesis method in Alchemy that permits continuous modulation of Position. In the example below, we use the Granular engine, but the same technique can be applied to the Additive and Spectral engines.

1. **Initialize Alchemy** by choosing the 'Clear' command in the Title bar's FILE menu.
2. **Load** (into Source A) a rhythmic or melodic sample that works well when looped. Try 'Factory' > 'Loops' > 'DrumLoops' > 'Cyborg-4bts.wav'. (The Load command imports this audio file quickly into the Granular engine. You could also Import the file, using any import mode, and the technique presented here would work fine.)
3. Now if you play and hold C3, the 'Cyborg-4bts' sample will **play in a looped fashion**. (Note that the Loop mode is already set to 'Continuous' by default.) If your host tempo happens to be 120 BPM, the loop will stay in tempo reasonably well — but suppose you want to sync the loop automatically to a different tempo set in your host.
4. Turn the Source A Stretch knob down to its minimum value of 0%. Now if you play notes you'll find that **playback is frozen at the very beginning of the sample**. (All you'll hear is a low rumble, because all of the individual grains are being drawn from the low bass sound that begins the sample.)
5. Next we'll **assign a modulator to the Position parameter** in order to control a playback path through the audio data. First we want the Position to increase smoothly so that the entire sample plays back from beginning to end, then we want the Position to jump immediately back to the beginning, and finally we want to repeat this path so that the sound loops. The right modulator for this job is **an LFO with a ramp up shape**. Click the Position knob in order to access its mod rack in the Modulation section. In the first mod rack slot, choose LFO 1, which brings the LFO 1 controls into view on the right-hand side of the Modulation section. Finally, choose 'RampUp' in the LFO Shape selection field and turn *off* the BIPOLAR button.
6. If you play and hold a note now, you'll find that the sample is looping as required, but the **playback speed** is much too fast. As its name suggests, the 'Cyborg-4bts' sample has a length of four beats. At its default rate, LFO 1 completes a cycle every beat. **Adjust the LFO Rate** to '4 beats'. Play and hold another note to confirm that the playback speed is now correct.
7. Finally, **adjust your host tempo** as you play additional notes to confirm that the loop is properly synchronized with the host tempo.

To summarize what's happening in this result: a tempo-synced LFO with a ramp up shape is controlling the Position knob, which causes playback to scan through the audio data in a forward direction every four beats. The result is a tempo-synced loop. The diagram below depicts these relationships.



Note that it's not actually necessary to 'freeze' the playback by setting Stretch to 0%, as we've done in step 4, before modulating Position. The LFO shape we've chosen (ramp up) is constantly changing, without any static/flat portions; this means that it always controls play position, which makes the value of Stretch irrelevant. (Try double-clicking Stretch to restore its default 100% value. Then play a note: the looping behavior remains the same.) However, if we were to use a different LFO shape with static/flat portions, then setting Stretch to a non-zero value would cause play position to be controlled by a combination of the LFO shape and the normal play behavior of the sample. (Try changing the LFO shape to 'RandHold' — the results may be more musically interesting if you also increase the LFO rate to '1/2 beats'. Then play notes and compare the results you get with Stretch set to 0% versus 100%.)

Morph

The **Morph X and Y controls** determine how Alchemy's four Sources interact. There two basic types of interaction.

- **XFade**. In an xfade (or 'crossfade'), sounds from all four Sources are played at once, and the X and Y knobs control the mix between them. This is equivalent to turning the Amp knobs in each Source up and down to get the desired mix.

  If you xfade from a Source with a high Coarse Tune setting to a Source with a low Coarse Tune setting, then the high Source will fade out as the low one fades in, and in the middle of the xfade you'll hear the high and low Sources simultaneously.

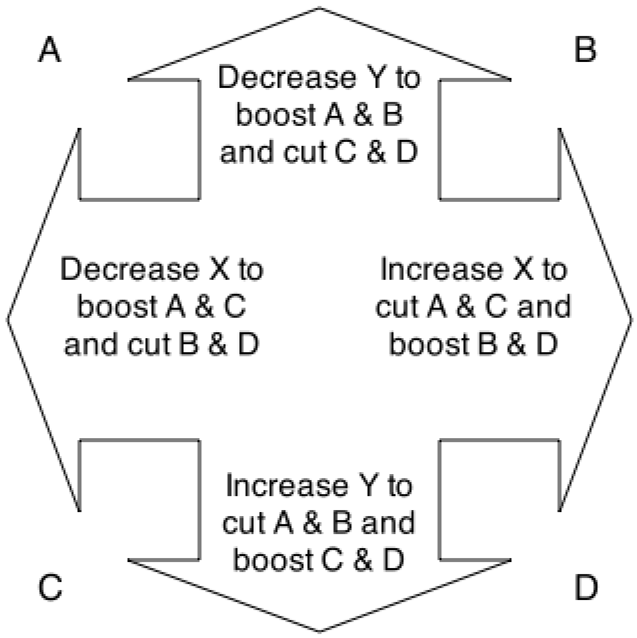
- **Morph**. In a morph, a single sound is generated, and the parameters of that sound are determined by interpolating between the settings of all four Sources.

  If you morph from a Source with a high Coarse Tune setting to a Source with a low Coarse Tune setting, then you'll hear a single sound during the entire morph, and its tuning will fall smoothly from the high value to the low one.

The **Morph pop-up menu** offers a variety of xfade and morph modes.

- **xfade linear** — The X knob crossfades from Source A to B to C to D as you increase it from 0% to 100%. (The Y knob does nothing in this mode.) This mode is useful for setting up crossfades based on Velocity or KeyFollow.
- **morph linear** — This mode is like xfade linear, but all the parameters of the sound are morphed. Regions of each source sound bounded by corresponding Warp Markers are time-aligned in the morph — the Source Edit (Main) page provides a fuller explanation.

- **xfade xy** — This is the **default morph/xfade mode**, and it crossfades between all four sources depending on the values of the X and Y knobs. X controls the mix levels of Source A & C versus B & D. Y controls the mix levels of Source A & B versus C & D.



- **morph xy** — This mode is like xfade xy, but all the parameters of the sound are morphed. Regions of each source sound bounded by corresponding Warp Markers are time-aligned in the morph — the Source Edit (Main) page provides a fuller explanation.

> ***Note that all of the remaining modes work similarly to morph xy, except that they provide control over the morph position of particular aspects of the sound. The settings you make in each of these more particular modes coexist with one another — for instance, if you first use the X and Y knobs in add xy mode (described below) to set values for the MorAddX and MorAddY parameters, then you can afterwards use the X and Y knobs in spec xy mode (described below) to adjust the MorSpecX and MorSpecY parameters without losing your MorAddX and MorAddY settings.***

- **time xy** — The X and Y knobs control the MorTimeX and MorTimeY parameters, which morph the timing of the sound, so if Source A has a short attack and Source B has a long attack, the length of the attack will vary as you change the X knob. In this mode, all the main Source parameters such as Amp, Pan, Tune, FilterMix, Position, and Stretch are morphed as well. (The attack portion of each sound is determined by the positioning of Warp Markers in the Source Editor's MAIN view.)

- **add xy** — The X and Y knobs control the MorAddX and MorAddY parameters, which morph the Additive element between the settings of Sources A, B, C, and D.

- **spec xy** — The X and Y knobs control the MorSpecX and MorSpecY parameters, which morph the Spectral element between the settings of Sources A, B, C, and D.
- **gran xy** — The X and Y knobs control the MorGranX and MorGranY parameters, which morph the Granular element between the settings of Sources A, B, C, and D.
- **filt xy** — The X and Y knobs control the MorFiltX and MorFiltY parameters, which morph the settings of the three Source Filter cutoff and resonance values between the settings of Sources A, B, C, and D. (The filter types and on/off status set in Source A remain in effect throughout the morph.)

> ***See the example at the bottom of this page for discussion of an easy way to work with several of these more particular morph modes simultaneously.***

What parameters participate in a morph?

While most parameters can be morphed, parameters that are set via a pop-up menu cannot. For example, the values of Source Filter Cutoff and Resonance knobs can be morphed, but the filter types cannot. Similarly, the Additive Pitch, Amp, and Pan knobs can be morphed, but the profiles associated with these knobs — such as 'Harmonic' and 'OddEven' — cannot. Whenever a parameter cannot be morphed, the setting in Source A remains in effect throughout the morph.

Where did the 'Morph Square' go?

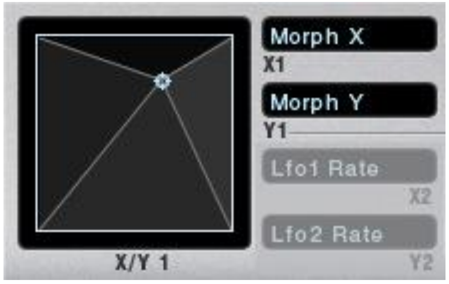
Users of Camel Audio's Cameleon 5000 additive synthesizer will already be familiar with the concept of morphing between four sources. In Cameleon 5000, these were called 'voices', and they were configured as the four corners of a Morph Square. Dragging a control point in this square gave users manual control over morphing, and complex automated morphs within the square could be created with the Morph Timeline.

In Alchemy, morphing between Sources A–D is controlled directly by the Morph X and Y knobs. But Alchemy offers equivalents to both the Morph Square and the Morph Timeline. Specifically, you can use one of the XY Pads in the Performance section like a **'Morph Square'** by assigning its X axis to modulate the Morph X knob and its Y axis to modulate the Morph Y knob. (See the example below for more details.) And you can use an XY MSEG in the Mod section like a **'Morph Timeline'** by assigning its X MSEG to modulate the Morph X knob and its Y MSEG to modulate the Morph Y knob. (See the example below for more details.)

Example: Basic additive and spectral morphing

- **Initialize Alchemy** by choosing the ‘Clear’ command in the Title bar’s FILE menu.
- **Import** a Factory Sample into each Source A–D, using ADD+SPEC mode. The following choices work well for the purposes of this example:
    - Source A: ‘Woodwinds’ > ‘Flute-Alto’ > ‘Flute-Alto-C3’
    - Source B: ‘Keys’ > ‘Piano-BabyG’ > ‘Piano-BabyG-C3’
    - Source C: ‘Keys’ > ‘EPiano-FM-Classic’ > ‘EPiano-FM-Classic-C3’
    - Source D: ‘Woodwinds’ > ‘Oboe-Vib’ > ‘OboeB-C3’
- Go to each Source Spectral sub-page and double-click the **High Pass** knob; this reduces the High Pass setting and makes the Spectral component of the sound more prominent (which is useful for the purposes of this example). Then return to the **Source All** sub-page, which brings the **Morph X** and **Y** knobs into view.
- Choose **‘morph xy’** from the Morph mode selection field. Now you can use the Morph X and Y knobs to morph between Sources A–D.

- Next, let’s assign an XY Pad (in the Perform section) to control the MorAllX and MorAllY parameters.
    - Set the Morph X knob to 0.0%; then right-click (control-click) it and choose ‘Add Modulation’ > ‘Perform’ > ‘XYPad1X’. Likewise, set the Morph Y knob to 0.0%; then right-click it and choose ‘Add Modulation’ > ‘Perform’ > ‘XYPad1Y’.
    - Now you can morph between Sources A–D by dragging the control point in XY Pad 1. Users of Camel Audio’s Cameleon 5000 synthesizer will recognize that this result replicates Cameleon 5000’s Morph Square in ‘All’ mode.



Example continued: Morphing different aspects of the sound

In the previous example — which is a starting point for this one — we replicated Cameleon 5000's Morph Square in 'All' mode. In Cameleon 5000, it is also possible to switch the Morph Square to 'Amplitude', 'Harmonics', or 'Noise' mode. Color-coded control points for each of these modes are superimposed on Cameleon 5000's Morph Square, and you can set set the position of each of these points separately by switching among the different modes. Alchemy's XY Pads are not designed to display multiple, color-coded control points. In this next example, we'll explore an easy way to have simultaneous, independent control over the morphing of different aspects of a sound in Alchemy.

- To begin, 'undo' the XY Pad assignment made in the final step of the previous example: right-click (control-click) XY Pad 1 and choose 'Delete Modulation X' > 'MorAllX'; then right-click XY Pad 2 and choose 'Delete Modulation Y' > 'MorAllY'.
- Next, choose 'time xy' from the Morph mode selection field. This gives you control over the MorTimeX and MorTimeY parameters and corresponds to Cameleon 5000's 'Amplitude' morph mode. Set the Morph X knob to 0.0%; then right-click (control-click) it and choose 'Add Modulation' > 'Perform' > 'Control1'. Likewise, set the Morph Y knob to 0.0%; then right-click it and choose 'Add Modulation' > 'Perform' > 'Control5'.
- Similarly, choose 'add xy' from the Morph mode selection field. This gives you control over the MorAddX and MorAddY parameters and corresponds to Cameleon 5000's 'Harmonics' morph mode. Set the Morph X knob to 0.0%; then right-click (control-click) it and choose 'Add Modulation' > 'Perform' > 'Control2'. Likewise, set the Morph Y knob to 0.0%; then right-click it and choose'Add Modulation' > 'Perform' > 'Control6'.
- Finally, choose 'spec xy' from the Morph mode selection field. This gives you control over the MorSpecX and MorSpecY parameters and corresponds to Cameleon 5000's 'Noise' morph mode. Set the Morph X knob to 0.0%; then right-click (control-click) it and choose 'Add Modulation' > 'Perform' > 'Control3'. Likewise, set the Morph Y knob to 0.0%; then right-click it and choose'Add Modulation' > 'Perform' > 'Control7'.
- This result gives you six Performance knobs that you can use to control morphing of the additive, spectral, and envelope-timing aspects of Sources A–D. (Try using MIDI learn to map these knobs to your controller hardware!)

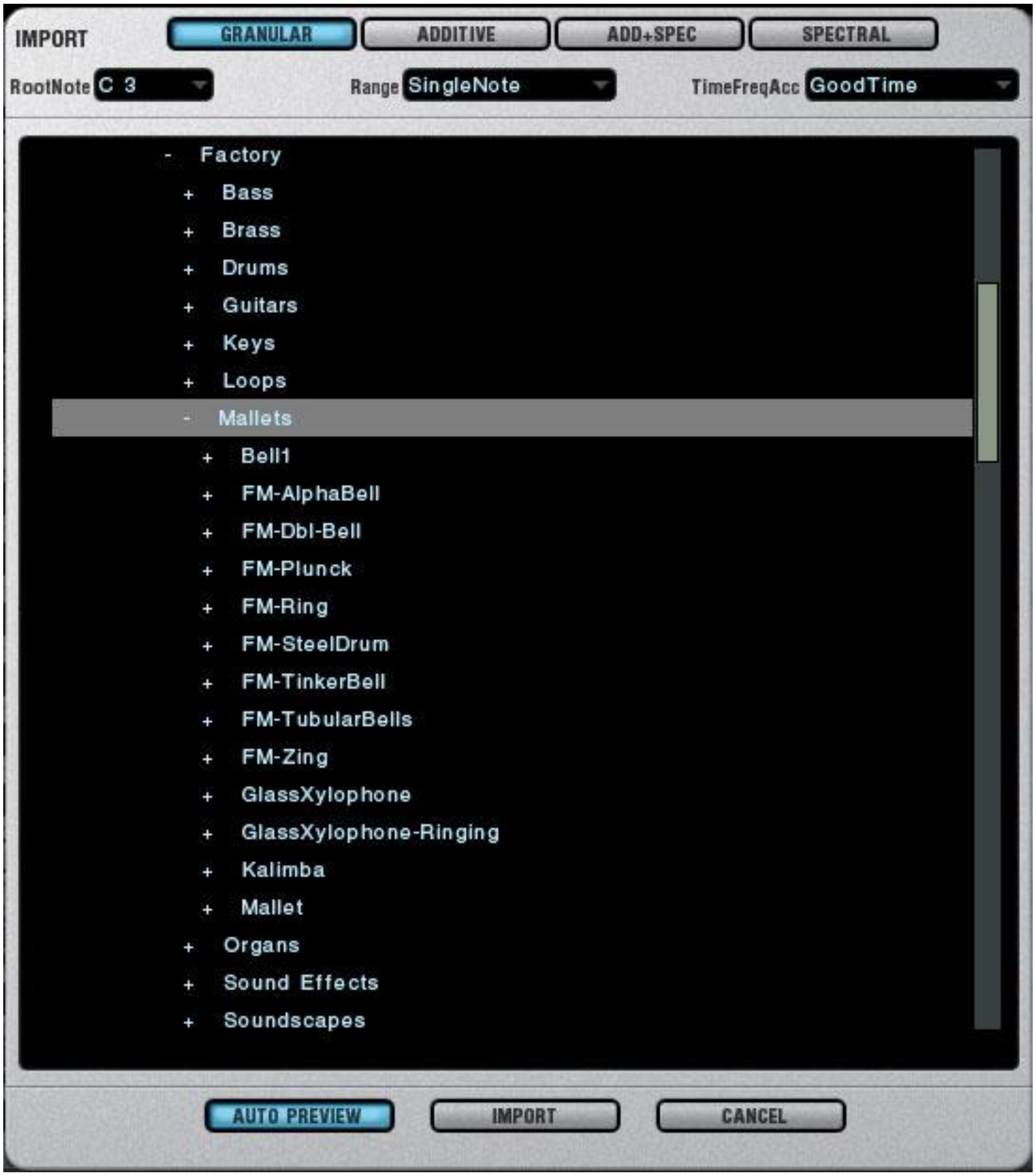
Example continued: MSEG-driven morphs

Let's take our example one step further and construct the equivalent of Cameleon 5000's 'Morph Timeline', which is a powerful tool for creating complex, rhythmic morphs. You can use the results of either of the preceding examples— 'basic additive and spectral morphing' or 'morphing different aspects' — as a starting point for the steps below.

- **Undo any Performance control assignments**: right-click (control-click) all the active Performance knobs and use the 'Delete Modulation' options to find and delete knob modulations; or right-click (control-click) the active XY Pad and use the 'Delete Modulation X' and 'Delete Modulation Y' options to find and delete pad modulations.
- If necessary, restore the morph mode to **'morph xy'**. Right-click (control-click) the **Morph X** knob and choose 'Add Modulation' > 'MSEG' > 'MSEG 1'; then right-click the **Morph Y** knob and choose 'Add Modulation' > 'MSEG' > 'MSEG 2'.
- Click the XY MSEG button at the top right of the Modulation section to bring the **XY MSEG controls** into view. Note that MSEG 1 and MSEG 2, which we have already assigned to modulate the morph knobs, both participate in the default XY MSEG configuration. (There is a dedicated page describing how to work with the XY MSEG controls; for this example, we'll simply load a preset into the default configuration.)
- **Load an XY MSEG preset**: click the FILE button, and choose 'Load' from the pop-up menu. From the factory 'MSEG' folder, choose 'Rhythmic-Simple' > 'RSquare 1_4X.mse'. The chosen file is loaded into MSEG 1, and the associated file 'RSquare 1_4Y.mse' is automatically loaded into MSEG 2 as well.
- Play a few held notes to confirm that the sound now morphs rhythmically between the settings of Sources A–D.
- You may find that Alchemy's faster envelope response leads to unwanted 'clicks' each time the morph position jumps to a different corner of the square. Let's make one refinement to eliminate any artifacts of this type. Click the Morph X knob to bring the MorAllX parameter's mod rack into view. (You'll see MSEG 1 is already assigned as a modulator.) Find the Smooth knob above the mod rack and raise it slightly, to about

1.0%. Likewise, click the Morph Y knob to bring the MorAllY parameter's mod rack into view, and again raise the Smooth knob to about 1.0%.

Import



Choosing ‘Import’ from the pop-up menu in a Source content field opens the **Import** window.

The central portion of the window provides a simple file browser in which you can select WAV, AIFF and SFZ format sound files for importing. The **AUTOPREV** button toggles automatic previewing on and off. With AUTOPREV activated, you can hear a preview of a sound file simply by clicking its name. Sound files can be mono or stereo, 8-, 16- or 32-bit, at any sample rate (although rates higher than 44.1 KHz are not recommended, since they don’t yield any significant improvements in quality).

You can import sound files to any of Alchemy's various elements: to granular, additive, or spectral, or to a combination of additive and spectral. To choose one of these import modes, use the GRANULAR, ADDITIVE, ADD+SPEC, and SPECTRAL buttons across the top of the Import window. Importing to the additive and/or spectral elements requires Alchemy to perform a special analysis of the file, which may take a moment.

A description of each import mode follows, with comments on the relevant options.

Granular

Importing to the granular element is good for drums loops, percussive sounds, and any sound to which you want to apply special granular effects. You can also use granular import and then switch the Granular element to its ordinary 'Sampler' mode, which is the best choice when you want to play a sound file efficiently in its original form, transposed across the keyboard.

- **RootNote** determines which MIDI note will play the sound file at its original transposition. If the filename has a pitch appended to it ('MyFile C2', 'YourFile-E3', 'OurFile_F#4'), Alchemy will automatically set the RootNote parameter to match.
- The **Range** and **TimeFreqAcc** parameters do not apply to Granular import.

> ***Note: If the sound quality is not what you expected, check to make sure the Granular element is switched to 'Sampler' mode, unless you intend to perform granular-specific manipulations such as time stretching/shuffling.***

Additive

Importing to the Additive element allows for the most detailed manipulation of sounds, and is especially good for sound files that represent single notes (rather than chords or more complex sounds and textures).

- Good results with additive importing depend on an appropriate choice of **RootNote**. If the filename has a pitch appended to it, Alchemy will automatically set the RootNote parameter to match; otherwise, it will attempt to determine a suitable RootNote based on a preliminary analysis of the waveform.

> ***Note: If a sound fails to import as well as you think it ought to, you may be able to get improved results by shifting the RootNote setting up or down an octave.***

- The **Range** setting is also significant. For sound files that represent single notes (rather than melodies, chords, or more complex sounds and textures), choose ‘SingleNote’. For most other sounds, choose ‘OneOctave’. Again, Alchemy will attempt to determine an appropriate setting based on a preliminary analysis of the waveform.
- **TimeFreqAcc** determines whether ‘time accuracy’ or ‘frequency accuracy’ is prioritised during the import process. In most cases you can leave this at the default (‘GoodTime’), but for some material different settings may produce better results. (For example, to analyse speech, you should probably choose ‘BestFreq’.)

Add+Spec

When a purely Additive import fails to capture the noisy components of a sound (such as the hammer strike of a piano or the breath noise of a flute), a combination of Additive and Spectral importing may give the best results.

- After an Add+Spec import, try deactivating the Source’s Additive element and listening to the Spectral element on its own. This will allow you to hear which parts of the sound failed to be reproduced by the Additive element, and should give you an idea what kinds of sounds are better-suited to the different analysis methods.
- By default, the Spectral component of an Add+Spec import is played with the Spectral element set to ‘Noise-Resynth’ mode. Playing the spectral data back in ‘Resynth’ mode produces a markedly different effect, which you may sometimes prefer.
- By default, the Spectral element’s internal highpass control is set relatively high in order to exclude frequencies that would compete with those produced by the Additive element. If you find the default contribution from the Spectral element to be too subtle, you can make it more prominent in many cases by setting the highpass control to a lower value.

Note: If a sound fails to import as well as you think it ought to, you may be able to get improved results by shifting the RootNote setting up or down an octave.

Spectral

Importing to the Spectral element allows effective manipulation of polyphonic sounds (chords, drum loops, and other complex sounds and textures that don’t analyse well via Additive).

- Set **RootNote** to determine which MIDI note will play the resynthesized sound at its original transposition. If the filename has a pitch appended to it, Alchemy will automatically set the RootNote parameter to match.
- The **TimeFreqAcc** parameter can also be adjusted to suit different types of source material (see details in the discussion of Additive import, above).
- The **Range** parameter does not apply to Spectral import.

'Import' versus 'Load'

The 'Load Audio' and 'Import Audio' commands available in the Source content field's pop-up menu both bring audio files into Alchemy.

'Load Audio' can be used load a WAV, AIFF, or SFZ file rapidly into the Granular element for playback in Granular or Sampler mode. You can also load files by clicking the Source content field's forward and back arrows.

'Import Audio' is for when you want Alchemy to analyze and resynthesize a sound file, or when you want to preview sounds before loading them. When a sound has been imported with additive or spectral analysis, the resynthesis data is stored in an '.aaz' file in the same folder where the preset is saved. The first part of the .aaz filename is the same as the preset name, ensuring that the preset and its associated .aaz files will be found side by side in an alphabetical directory.

> ***Note: It is recommended that you avoid renaming .aaz files externally (i.e. from your operating system); Alchemy may fail to locate and load these files unless their original names are preserved. While renaming a whole preset externally is less risky, it may lead to duplication of .aaz files to reflect the new name. In general, it's best to rename presets by using Alchemy's SAVEAS function. Once a preset is successfully saved under its new name, you can safely discard the old preset (*.acp) file.***

Supported SFZ opcodes

Alchemy is capable of importing multi-zone sample data in SFZ format. SFZ is a non-proprietary file format (described at http://www.cakewalk.com/DevXchange/sfz.asp). Alchemy recognizes the **<region>** and **<group>** headers and the following opcodes:

sample
pitch_keycenter

lokey
hikey
lovel
hivel
loop_mode
cutoff
fil_veltrack

You can create your own SFZ files for use in Alchemy by using various third-party software tools. SFZ definition files are simple textfiles, so another approach to assembling your own SFZ files is to place a copy of an existing definition file in a folder along with the new samples you want to use, and then to edit the definition file — in a text editor such as NotePad (PC) or TextEdit (Mac) — so that it references your new samples.

Additive

A brief introduction to additive synthesis…

Additive synthesis produces complex sounds by summing together a number of simpler ones, known as **partials**. Typically, each partial is a pure sine wave tuned to be a multiple of some fundamental frequency *f*, so that you have partials at *f*, 2*f*, 3*f*, and so on. Additive synthesis puts the mathematician Joseph Fourier's theory, which states that every static waveform can be decomposed into a sum of sine-wave partials, into practice.

Of course most musically useful sounds involve waveforms that **change over time**; producing these sounds with additive synthesis requires changing the level of each partial accordingly. Another important additive technique involves tuning partials higher or lower than their harmonic positions. These **inharmonic partials** are characteristic of the patterns produced when stiff materials such as metal and glass vibrate.

Two features make the additive engine in Alchemy so powerful. First, it offers an unusually **large number of partials** - as many as 600 in each of its four Sources. And second, the amplitude, pitch, and pan of each partial can be specified across an **unlimited number of breakpoints**. That adds up to a potentially huge amount of data, but Alchemy can manage the details for you thanks to its **resynthesis** capabilities.

Resynthesis is a two-step process. First, an audio file (in a standard format such as WAV or AIFF) is **analyzed** to determine its partials and their behavior over time. And second, a replica of the original sound is **synthesized** using the results of the analysis.

Resynthesis in Alchemy produces results that are remarkably close to the sound of the original audio file. But then the real fun begins: because you now have independent control over the amplitude, pitch, pan, and phase of each partial, you can manipulate sounds in ways that would be impossible to achieve with a conventional sampler.

Alchemy enables you to perform these manipulations in real-time using a variety of modulation possibilities, described below. For precision work, you can also make use of Alchemy's dedicated Additive Editor, which allows you to work at varying levels of detail, right down to individual partial values and breakpoints.

ADD and VA mode buttons

The Additive element in each of Alchemy's four Sources can operate in one of two modes. The ADD button on the left of the Additive sub-page puts the element in **Additive** mode, while the adjacent VA button puts it in **Virtual Analog** mode. (Turning on one mode turns the other off.) Importing additive data, or creating it from scratch in the Additive Editor, automatically sets the mode to Additive.

Controls on the Additive sub-page apply to both VA and Additive mode. The descriptions below explain how these controls work in Additive mode. The effect of the same controls in VA mode is explained on the Virtual Analog (VA) page.

Using the Additive sub-page controls in ADD mode

- Wave pop-up menu. Selects the **waveform** used to play each partial. In ADD mode, the 'Sine' waveform in the 'Basic' category is nearly always preferred, as it allows more accurate resynthesis and requires considerably less CPU load. When the total number of partials is very small, however, other choices of waveform may give you richer and more interesting results.

- **Vol**. Adjusts the **volume** of the Additive element, independent of other elements in the same source.

- **Sym**. Controls the **symmetry** of the waveform used to play each partial. At the default value of 50%, this control has no effect. At higher or lower values, one half of the waveform is stretched while the other half is compressed. Modulating this parameter produces pulsewidth modulation. To minimize CPU usage, however, leave this control at 50% and avoid modulating it.

- **PVar**. Use this control to reduce the amount of **pitch variation** present in the additive data. At 100%, the original pitch variation is applied in full; at 0%, the pitch of each partial is fixed at its harmonic value. (If you're familiar with Cameleon 5000, keep in mind that 'PVar' in Alchemy is equivalent to 'Harm' in Cameleon 5000.)

  Note that pitch offsets resulting from the Pitch knob (described below) are not affected by the PVar setting.

- **NOsc**. Use this control to reduce the **number of oscillators** used to play the additive data (and hence the number of higher partials that are heard). Using fewer oscillators conserves processing power, and you can often reduce NOsc without noticeably changing the resulting sound. More drastic reductions in the NOsc value produce an effect comparable to lowpass filtering. (Try modulating this parameter with an AHDSR.)

Note that the additive engine processes partials in groups of four, so setting NOsc to a multiple of four gives you the best trade-off between high-end detail and CPU efficiency.

- **Pitch** knob and pop-up menu. The Pitch knob applies a variable offset to each partial pitch. The nature of this offset depends on the setting chosen in the Pitch pop-up menu directly below the knob:
    - 'Harmonic' - this setting is the default in ADD mode and is the most useful choice in most situations. When Pitch = 0%, all partials are tuned precisely in unison at the fundamental frequency. When Pitch = 50%, all partials have their normal harmonic tuning. When Pitch = 100%, the partial tunings are stretched significantly. (If you're interested in the math behind this stretching: partial *n* is tuned to the (2*n* – 1)th harmonic.)
    - 'Unison' - this setting is mainly intended for VA mode and has limited use in ADD mode. (See the description on the Virtual Analog (VA) page for details.)
- **Amp** knob and pop-up menu. The Amp knob applies a variable offset to each partial pitch. The nature of this offset depends on the setting chosen in the Amp pop-up menu directly below the knob:
    - 'Brightness'- set Amp above 50% to boost the brighter (higher) partials, or below 50% to boost the darker (lower) ones.
    - 'Fifths' - set Amp above 50% to boost the partials that are a fifth, plus one or more octaves, above the fundamental, or below 50% to boost the other partials.
    - 'FiveMul' - set Amp above 50% to boost every fifth partial (5, 10, 15, 20, …) and cut the others, or below 50% to cut every fifth partial and boost the others.
    - 'Fundamental' - set Amp above 50% to boost the fundamental and cut the other partials, or below 50% to cut the fundamental and boost the other partials.
    - 'Octaves' - set Amp above 50% to boost the partials that are one or more octaves above the fundamental, or below 50% to boost the other partials.
    - 'OddEven' - set Amp above 50% to boost the even partials and cut the odd ones, or below 50% to do the reverse. Boosting the odd partials creates a 'hollow' tone resembling the sound of a square wave.
- **Pan** knob and pop-up menu. The Pan knob applies a variable offset to each partial's stereo position. At 50%, all partials are centered. Values lower than 50% pan some

partials left and other partials right; values greater than 50% pan partials in the opposite direction. The nature of these offsets depends on the setting chosen in the Pan pop-up menu directly below the knob:

- ‘Brightness’ - pans darker partials to one side and brighter partials to the other.
- ‘OddEven’ - pans odd partials to one side and brighter partials to the other.
- ‘Unison’ - pans partials evenly across the stereo field.

Note that you won’t hear the effect of Pan unless the Source’s Stereo button is engaged.

Example: Manipulating a resynthesized voice

- Initialize Alchemy using by selecting FILE > ‘Clear’ in the Title Bar.
- Click the ‘A’ button to view the Source A sub-page and choose ‘Import Audio’.
- In the left-hand half of the Import dialog, select the sample ‘Factory’ > ‘Vocals’ > ‘Caroline-CamelAudio.wav’; then set the import mode to ADD+SPEC at the top right, and click IMPORT at the bottom right.
- Now you can play a resynthesized version of the ‘Caroline-CamelAudio’ sample — play F-sharp 3 to hear the original pitch.
- Try the following manipulations:
    - Reduce NOsc from 32 to about 4 and note how the voice becomes darker and less distinct. (Reset NOsc to 32.)
    - Try stretching and compressing the tuning of the partials by adjusting the Pitch knob. Because Caroline sings clear pitches, the original spectrum is harmonic; when stretched or compressed, the spectrum becomes inharmonic at most settings, giving it a glassy or metallic quality. A setting of 25% gives the effect of a deep male voice. (Reset Pitch to 50%.)
    - Explore the Amp and Pan knobs as well. (Turn on the Source STEREO button to hear the effect of Pan.)
- Finally, let’s turn ‘Caroline-CamelAudio’ into a pad sound.
    - Set PVar to 0% in order to provide a steady pitch.
    - Slow down the playback by setting Stretch = 30% at the top of the Source A panel. Turn on the Source STEREO button.

- Now set Pitch = 25% (using the ‘Harmonic’ profile), Amp = 0% (using the ‘OddEven’ profile), and Pan = 100% (using the ‘Unison’ profile).
- Refine this result by adjusting the rest of the signal path. Explore the Main Filters. Try adding a ModFX or Delay module in the Effects rack.

Advanced: Creating your own Pitch, Amp, and Pan profiles

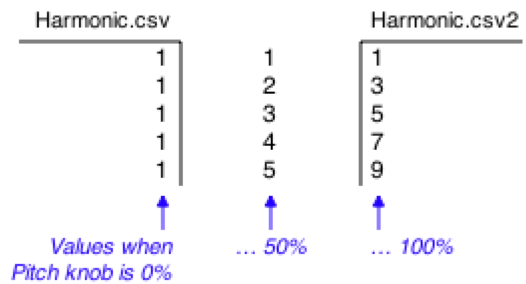
The preset profiles for Oscillator Pitch (Unison, Harmonic), Amp (Brightness, OddEven, etc.), and Pan (Unison, OddEven, etc.) are defined by special files stored in Alchemy/Libraries, where you will find subfolders named OscillatorPitch, OscillatorAmp, and OscillatorPan. These profile definitions are formatted as standard CSV files, which makes it easy to create your own definitions by using a text editor or spreadsheet, or by writing your own computer program that stores its output in a text file.

Each profile definition requires a pair of files with identical base names and different extensions: the profile corresponding to a knob value of 0% has the extension .csv, while the profile corresponding to a knob value of 100% has the extension .csv2. You may find it informative to study an existing pair, such as ‘Harmonic.csv’ and ‘Harmonic.csv2’ in the OscillatorPitch folder. Or read on for a more detailed explanation.

Each row of a profile definition holds just one value, corresponding to the pitch, amp, or pan value of a single partial. The first row applies to the first partial, the second row to the second partial, and so on. Since Alchemy generates up to 600 partials, an exhaustive profile definition should consist of 600 rows. If you are creating a new profile for a specific purpose, and you know you won’t use more than, say, 60 partials, then there’s no need to continue the profile definition beyond 60 rows. (If you use more oscillators than there are values in the profile definition, the surplus oscillators retain their previously set values.)

In a profile definition, pitch values are expressed as multiples of the fundamental frequency. (So the values ‘1, 1, 1, …’ represent an exact unison; the values ‘1, 0.99, 1.01, …’ represent a detuned unison; the values ‘1, 2, 3, …’ represent the standard harmonic series; and so on.) Amp and Pan values are expressed on a scale from 0 to 1. (In the case of Pan values, this means from fully left to fully right in the stereo field).

Alchemy determines actual oscillator values by interpolating between the .csv and the .csv2 files, based on the position of the Pitch, Amp, or Pan knob. The details in the case of the ‘Harmonic’ Pitch profile are pictured below.



Let's create a new Amp profile that will enable us to cut or boost the levels of the [prime-number]() partials relative to the others. For simplicity, we'll extend our profile only through the first 60 partials; constructing files of this length by hand in a text editor is a manageable task. When the Amp profile is set to our new 'Primes' definition, turning the Amp knob fully left should set the amplitudes of all prime-number partials to 0, and all nonprime-number partials to 1 (this tells us how 'Primes.csv' should be constructed), and turning the Amp knob fully right should do the reverse (this tells us how 'Primes.csv2' should be constructed).

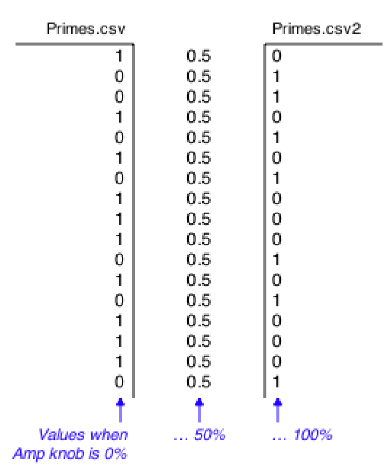
The prime numbers up through 60 are: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59. So creating the 'Primes' profile involves the following steps:

1. Create a plain textfile with one value per line, that value being '0' for the prime-number lines (2nd, 3rd, 5th, 7th, 11th, …) and '1' for the nonprime-number lines (1st, 4th, 6th, 8th, 9th, …); then save this file in Alchemy/Libraries/OscillatorAmp with the name 'Primes.csv'.

2. Create a second plain textfile with one value per line, that value being '1' for the prime-number lines and '0' for the nonprime-number lines; then save this file in Alchemy/Libraries/OscillatorAmp with the name 'Primes.csv2'.

3. The next time you load Alchemy in your host, the 'Primes' Amp profile should be listed along with the factory profiles. Give it a try!

Here's an illustration of how the 'Primes' profile works (going as far as the 17th partial):



Virtual Analog (VA)

The **VA** button on each Source's Additive element sub-page gives you access to Alchemy's 'Virtual Analog' mode. In this mode, you can use the additive engine as a bank of raw oscillators such as you might find in a classic analog synthesizer.

The **controls** in VA mode are the same as in ADD mode. The Additive page gives you detailed descriptions of each control. The discussion below focuses on using these controls to perform various analog-synthesis tasks.

When you **initialize** Alchemy, using the 'Clear' command in the Title bar's FILE menu, the Source A Additive element is turned on and switched into VA mode. By default, a single oscillator is active and the Saw wave is selected.

VA techniques in Alchemy

To select the oscillator waveform, use the **Wave** pop-up menu. The 'Basic' category offers Saw, Sine, Square, and Triangle waves. A large variety of more specialized waves is also provided. You can browse through the available waveforms using the forward and back buttons, or make a direct selection from the pop-up menu.

To adjust the pulsewidth, use the **Symmetry** control. A setting of 50% preserves the original waveform, while higher or lower settings stretch one half of the wave and compress the other half. Starting with a Square wave gives you a classic variable-width pulse wave, but you can adjust the symmetry of every waveform in Alchemy. Symmetry values as low as 5% and as high as 95% are typically useful; at the most extreme settings, the sound may become very thin or unstable.

For pulsewidth modulation (PWM), **modulate the Symmetry parameter** with an LFO, AHDSR, or other modulator.

To control oscillator phase, use the **PVar** knob. Values from 0% to 99.9% set a consistent starting position for the oscillator. When PVar = 100%, the starting phase of the waveform varies randomly per note.

> ***Note that the PVar knob has a different meaning in ADD versus VA mode. In VA mode, it provides control over phase. In ADD mode, it controls the depth of response to partial pitch variations in the additive data.***

Set the number of oscillators using the **NOsc** knob. (The Pitch knob, described below, provides unison with variable detuning.) Alchemy lets you stack up to 600 oscillators per source (!) — but the high end of this range is really intended to provide sufficient sinewave partials for detailed additive resynthesis. In VA mode, typical settings of NOsc range from 1 for single-oscillator sounds, up to 20 or so for 'supersaw' and related effects. Higher settings of NOsc consume more CPU.

To control the amount of detuning (with NOsc = 2 or more), choose **'Unison'** in the **Pitch** pop-up menu and adjust the **Pitch** knob. A value of 0% tunes the oscillators to a perfect unison, while 100% provides the maximum amount of detuning.

To spread unison oscillators across the stereo field, choose **'Unison'** in the **Pan** pop-up menu and adjust the **Pan** knob. A value of 50% keeps all the oscillators centered, while higher or lower values distribute the oscillators from left to right.

> ***Note that you won't hear the effect of Pan unless the Source's Stereo button is engaged.***

For white and filtered noise, use the Spectral element in WHITENOISE mode. (See the Spectral page for details.)

Example: Creating a 'supersaw' effect

- **Initialize Alchemy** by choosing the 'Clear' command in the Title bar's FILE menu. This puts Source A into VA mode.
- If the Source A **Additive sub-page** is not already in view, click the A button and then click the Source A ADDITIVE button.
- The **Saw wave** should already be selected in the Wave pop-up menu.
- Increase **NOsc** to about 10. You may also wish to increase the value of **Pitch** above 50% for more extreme detuning.
- Finally, try a couple of refinements:

- For **modwheel control of the filter cutoff**, go to the Perform sub-page and choose ‘Auto Assign All’ from the FILE menu at the far right. Filter cutoff is now mapped to Perform Control 3, so choose ‘Control3’ in the ModWheel pop-up menu.
- Experiment with **various effects modules** in the Effects rack. (Compressor and Camel Reverb are good ones to try.)

Spectral

A brief introduction to spectral synthesis…

Spectral synthesis and additive synthesis represent two different ways of creating complex sounds by summing together a number of simpler components. In additive synthesis (which you can read about here), each component is a sine wave partial tuned in relation to a harmonic series. In spectral synthesis, the audible spectrum is divided into a large number of ‘spectral bins’, and different sounds are represented as different distributions of energy across these bins. Normally, Alchemy plays spectral data by summing together a number of sine waves centered in each bin; it can also fill each bin with filtered noise.

> ***This process uses a model known as a phase vocoder. We use the term ‘spectral synthesis’ to avoid confusion: Alchemy does not provide a conventional ‘vocoder’ and is not designed to impose the spectral properties of one signal on another in real time.***

Alchemy lets you **resynthesize** existing audio files (WAV or AIFF) using spectral techniques as well as additive ones. Your choice may depend on the type of material: additive resynthesis is often preferred for sounds that project a single clear pitch, while spectral resynthesis is well suited to chords and noisy sounds. Alchemy can also perform resynthesis using a *combination* of the two techniques, which is useful for sounds that combine a clear pitch with a noisy component (e.g. the hammer strike of a piano, or the breath noise of a flute).

See the Import page for instructions on how to perform resynthesis.

Spectral resynthesis allows you to **manipulate your source material** in unique ways. You can switch between resynthesis using sine waves and resynthesis using filtered noise — this can transform speaking into whispering. And for detailed work, you can edit spectral data by applying various painting tools in Alchemy’s dedicated Spectral Editor.

Unlike the other elements in Alchemy, the Spectral element generates a **mono (rather than stereo) signal**, so the Source STEREO button has no effect on the Spectral element signal. If you import a stereo file using spectral resynthesis, the left channel will be used and the right channel will be discarded.

Modes of operation

The Spectral element in each source can operate in one of three different modes, selectable via the three Spectral Mode buttons at the left of the Spectral sub-page:

- **WHITENOISE**. Produces steady white noise.
- **RESYNTH**. Plays the current spectral data (from resynthesis or painted directly into the Spectral Editor) by summing sine waves.
- **NOISE-RESYNTH**. Plays the current spectral data (from resynthesis or painted directly into the Spectral Editor) by summing bands of filtered noise. The result is still more or less 'pitched', but with a distinctive noisy/phasey quality.

Controls

- **KEYTRACK** button. Determines whether or not the pitch of the Spectral element tracks the keyboard in RESYNTH and NOISE-RESYNTH modes. For keytracking to occur, the main Source KTRACK field must also be set to 'Key+PBend' or 'Key'.
- **Volume**. Adjusts the volume of the Spectral element, independent of other elements in the same source.
- **High Pass**. Sets the frequency below which spectral bands are eliminated (acts like a steep highpass filter).
- **Low Pass**. Sets the frequency above which spectral bands are eliminated (acts like a steep lowpass filter).

Granular

A brief introduction to granular synthesis…

Granular synthesis represents continuous sound as a stream of 'grains', or tiny pieces of sound. The effects that result from this technique depend on a variety of factors: the duration and amplitude envelope of each grain, the degree of overlap or separation from one grain to the next, and of course the internal details of each piece of sound, such as pitch and pan.

Alchemy **generates grains** by extracting 2- to 230-millisecond pieces from an existing sample (e.g. a WAV or AIFF audio file). The amplitude of each grain is shaped by a simple **window**, and **modifications** may be applied to its pitch and pan before it is sent to the output stream.

If each new grain is extracted from slightly further into the sample than its predecessor, then the resulting stream of grains will essentially 'put the pieces of sound back together' in a form that resembles the original source material. But a large number of interesting manipulations is also possible:

- **Time-stretching**. Grains can be sent out at a faster or slower rate than they are found in the original sample, causing it to play faster or slower without changing its pitch. You can even 'freeze' a sample at a certain point by extracting many grains from this one point.
- **Pitch-shifting**. Modifications to the pitch of each grain allow you to vary a sample's pitch without affecting its timing. By modulating the pitch and/or pan of each grain, you can also create effects such as spaciousness and blurring.
- You can **scramble** the order in which grains are played back to produce effects ranging from mild fuzziness to extreme mangling.

The Position and Stretch controls on the main Source sub-page permit a variety of time-stretching and -shuffling effects. (For details, see the Manual's Source page, especially the example involving tempo-synced loops at the bottom of that page.) Other effects can be created by manipulating the Granular sub-page controls described below.

GRANULAR and SAMPLER mode buttons

The Granular element in each of Alchemy's four Sources provides two different types of sample playback, which you can select between using the GRANULAR and SAMPLER buttons on the left of the Granular sub-page. **Sampler mode** provides conventional sample playback, described in a bit more detail towards the bottom of this page. **Granular mode** provides granular sample playback and allows for a much wider range of manipulations using the Granular sup-page controls.

Granular sub-page controls

In addition to the controls listed below, granular-mode playback is also affected by different Loop modes and by the settings (and modulations) of the Position and Stretch knobs at the top of each Source sub-page.

- **Volume**. Adjusts the volume of the Granular element, independent of other elements in the same source. Unlike the remaining controls, which apply only in Granular mode, **Volume** applies in Sampler mode as well.
- **Size**. Adjusts the duration of each grain from 2 msec to 230 msec.
- **Density**. Controls the number of potentially overlapping grains from 1 (no overlap) to 10.

The key to using the Size and Density parameters effectively is to understand how they interact. When Density = 1, it's useful to imagine a single machine firing one grain at a time into the output stream. As soon as one grain finishes, the next one fires, so if Size = 100 msec, there's a new grain every 100 msec. Increasing Density to 2 adds a second 'machine' that fires its grains in between those of the first — so now there's a new grain every 50 msec, and the grains from machine 1 overlap those from machine 2. Further increases in Density cause new grains to fire more frequently and overlap more heavily.

Setting Size around 100 msec and Density around 5 grains often works well for smooth sounds without sharp transients, like pads. Setting Size around 40 msec and Density around 2 grains often works well for drums and other sounds featuring sharp transients. Very small values of Size tend to produce a 'buzz' that masks the original pitch of the sample, while very large values of Size tend to have a 'rough and choppy' effect, but you can counteract both tendencies by increasing the Density. Finally, it takes more processing power to generate a larger number of overlapping grains, so you can save CPU by decreasing Density.

- **Window** pop-up menu. Determines the envelope shape that is applied to each grain. Trial-and-error is really your best guide. Bear in mind that the effect of different Window choices is often quite subtle, especially when you avoid extremely high or low settings of Size and/or Density.
- **RTime**. Applies a small random offset to the position in the sample from which each grain is extracted. (This is equivalent to modulating the Source Position knob with a RandGlide LFO and a low modulation Depth.) The default value of RTime is 3%, rather than 0%, because a small amount of randomization helps to smooth the output of the Granular element.
- **RPan**. Applies a random offset to the stereo position of each grain. (This is equivalent to modulating the Source Pan knob with a RandGlide LFO and a low modulation Depth.) You won't hear the effect of RPan unless the Source's Stereo button is engaged.

Note that modulations applied to the Granular element update with each new grain. This means for instance that modulating the Source Coarse Tune parameter with an LFO causes the stream of grains to rise and fall in pitch, but it does not create pitch sweeps within each grain. If granular Size is large and granular Density is low, modulations applied to Source parameters such as pitch may sound 'stepped' when you would expect them to sound smooth.

Using Sampler mode

Alchemy's Granular element can be switched to SAMPLER mode for conventional sample playback. In this mode, samples are played continuously rather than being split into grains. Transposing a sample upwards speeds it up, transposing it downwards slows it down. Sampler mode uses less CPU than granular mode, and imposes less coloration on the sound. The trade-off: granular manipulations of the sound are not available in sampler mode.

When Alchemy's Granular element is switched to sampler mode, the Source **Position** control determines the sample start point. You can modulate this parameter (e.g. with Velocity), and the modulation value at the moment a note begins will produce a corresponding offset in the sample start position. But changes in the modulation value are ignored as the note continues to play. The Source **Stretch** control likewise has no effect in sampler mode. Among the Granular sub-page controls, only the **Volume** control remains active in sampler mode; the other Granular sub-page controls work exclusively in granular mode.

Source Edit (Main)

The **Source Edit** page is accessed by clicking the **EDIT** button on any of the Source sub-pages (A, B, C, or D). It allows access to some of Alchemy's more specialised features.

On the left-hand side of the Source Edit page is a series of general controls.

- The **A**, **B**, **C**, and **D** buttons select a source for detailed editing. The illuminated button corresponds to the currently selected source.
- The **EDIT** button remains illuminated while the Source Edit page is in view. To **exit** this page and return to the current Source sub-page, click the EDIT button.
- Several controls from the current Source sub-page are accessible here as well:
    - The **ON** button toggles the source on (when lit) and off (when unlit).
    - The **Source Content** selection field identifies the current source data (e.g. a file you have imported); click this field for access to a pop-up menu with a variety of options for working with Source content and settings. (See the Source page for a description of each of these options.)
    - Additional controls let you adjust the **Loop** and **Keytrack** modes and toggle the **Stereo** setting. (Again, see the Source page for a description of each of these controls.)

- The lower left-hand side of the Source Edit page lets you configure a Source with **multiple Zones**, each with a particular **key range** and **velocity range**. (Multiple zones are created automatically whenever you load/import a multi-sample SFZ file.)
    - The **ZONE** selection field displays the number of the currently active zone. The active zone is the one whose range and warp-marker details are currently displayed and editable. To view and edit range and warp-marker details for a different zone, click the zone selection field and choose the desired zone number from the pop-up menu.
    - The **Zone Name** field displays the currently active zone's name, which cannot be edited.
    - **Root Key** determines which MIDI note will cause the zone sample data to play at its original pitch.
    - **Low Key** and **High Key** set the boundaries of the key range across which the active zone will play. For example, if you set High Key to E4, then the active zone will not play in response to MIDI notes from F4 upwards. (Note that the key ranges of different zones can overlap, in which case more than one zone at a time will play in response to MIDI notes in the overlapping region.)
    - **Low Vel** and **High Vel** set the boundaries of the velocity range across which the active zone will play. For example, if you set Low Vel to 64. then the active zone will not play in response to MIDI notes with velocities less than 64.

The right-hand side of the Source Edit page contains a window for detailed graphical editing of Source data. Buttons across the top give you a choice between three edit modes:

- **MAIN** — for working with the Source **loop** and **warp markers**. See below for details.
- **ADDITIVE** — for editing additive data or creating it from scratch. See the Additive Editor page for details.
- **SPECTRAL** — for editing spectral data or creating it from scratch. See the Spectral Editor page for details.

Using the Main Editor

Once audio data is loaded/imported into a Source, the Main Editor view provides an overview of the sound's natural amplitude envelope. A series of **warp markers** is superimposed on this overview. Each marker is depicted by a vertical line with a labelled handle that you can click and drag.

What do warp markers do? When you Morph from a sound with a fast attack to one with a slow attack, using *morph xy* or *time xy* mode, Alchemy smoothly adjusts the attack time according to the morph position. Warp markers define the boundaries of each sound's attack portion — and more generally, **warp markers define the boundaries of a series of time-aligned segments** when two or more sounds are morphed together. Furthermore, two of the warp markers define the **loop start** and **loop end** points of each sound.

Whenever you import additive, spectral, or granular data, Alchemy **automatically sets warp markers** at five positions: (1) the **very start** of the sound; (2) the end of the **attack** portion; (3) the **loop start** point; (4) the **loop end** point; (5) the **very end** of the sound.

> ***Note that the effect of the loop start and end points depends on a Source's Loop mode. See the <u>Source</u> page for more information on Loop modes.***

- To **move** a warp marker, drag its handle. (You can drag the loop start and loop end markers just like any other warp markers.)

- To redefine which warp markers serve as the **loop start** and **loop end** points, right-click (control-click) a marker and choose ‘Loop Start’ or ‘Loop End’ from the contextual menu.

  Note that you cannot position the loop start later than the loop end. You also cannot set the same warp marker to serve as booth loop start and loop end. However, you can set adjacent markers to serve as loop start and loop end and then drag them to the same position. This creates a ‘sustain’ point rather than an extended loop region.

- To **insert** a new warp marker, double-click at the desired position.

- To **delete** an existing warp marker, double-click its handle.

  If you plan to experiment with the warp marker settings, it’s a good idea to Save (or Copy) the Source, using commands from the pop-up menu in the source content field at the top left of the editor page (next to the ON switch). That way if you’re not happy with the new settings you can restore the original ones by Loading the Saved source (or Pasting the Copied source).

A FILE pop-up menu, accessible from the **FILE** button directly above the editor view, offers commands to **Mark 8 = slices**, **Mark 16 = slices**, and **Mark 32 = slices**. Each of these creates the designated number of warp markers and positions them at equal intervals across the entire sound.

Warp marker tips

One common reason for working with warp markers is to adjust the **loop start** and **loop end** markers. This can be useful for any sound that you want to play in a looped fashion. See the description of Loop modes on the Source page for more details.

Adjusting the **remaining warp markers** has no effect on a single source. But if you are morphing between two or more sounds, these warp markers give you precise control over the time alignment of different portions of each sound.

- If your sounds are **single notes**, then the placement of the five ***default*** warp markers usually gives good results. But you may wish to experiment with the placement of ***marker 2***, so that it’s at the point where it sounds like the end of the attack for each sound. This can produce a more convincing morph between sounds with contrasting attack qualities.

- If your sounds consist of **multiple events** — e.g. musical phrases, drum loops, a spoken sentence — you may find it useful to create ***additional*** warp markers. If you are morphing between two voices speaking the same sentence, placing a ***warp marker at***

the start of each word will help to preserve the integrity of each word during the morph. If you are morphing between musical phrases or drum loops with different ‘grooves’ or timing nuances, placing a ***warp marker on every beat, half beat, or quarter beat***, will help to ensure a smoother morph. (The **‘Mark 8/16/32 = slices’** commands give you useful starting points.)

Additive Editor

Once you have accessed the Source Edit page, clicking the ADDITIVE button at the top of the right-hand half of the page will bring Alchemy's Additive Editor into view.

Overview

The Additive Editor serves two purposes. First, it allows detailed editing of additive resynthesis data. And second, it allows you to design sounds from the ground up by creating the additive data yourself. In both cases, potentially large amounts of data are involved. Alchemy uses special graphical representations of the data in order to make these tasks manageable.

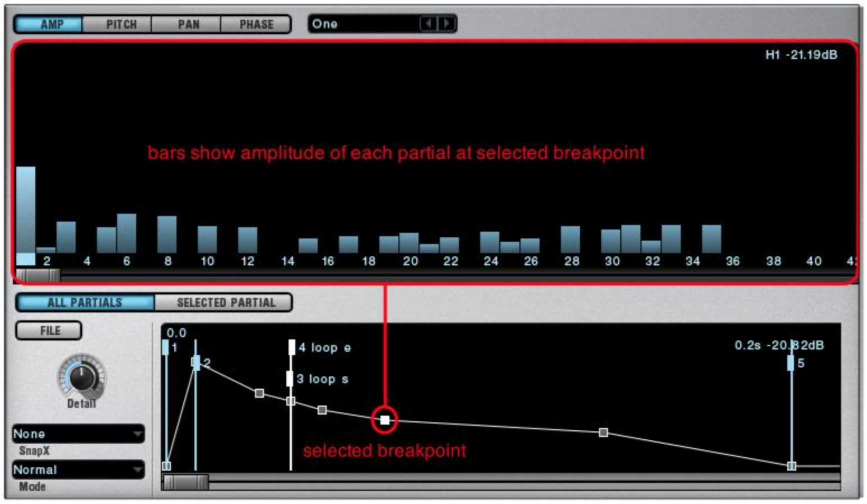
As explained on the main Additive page (see 'A brief introduction'), additive synthesis represents each sound as a sum of individual partials. The additive data describes each partial in terms of four parameters — amp, pitch, pan, and phase — each of which changes over time. **You can think of the additive data, therefore, as a series of 'snapshots', each of which captures the amp/pitch/pan/phase of every partial at a particular point in time.** In between snapshots, each parameter updates smoothly towards its next snapshot value. When played in succession and with the right timing, the whole series of snapshots describes a potentially complex and continuously evolving sound.

Note: Phase is not really an independent parameter. A partial's phase at any moment in a sound's evolution is determined by the phase at the beginning of the sound and by the (possibly changing) pitch of the partial. Since pitch information is captured in each snapshot, phase information doesn't have to be. The phase of each partial is only specified at the very beginning of the sound.

Breakpoints and partial bars

Alchemy presents the additive data using a pair of graphical displays. The first of these displays is a **breakpoint envelope** located in the bottom half of the Additive Editor view. Each breakpoint in this display corresponds to one of the 'snapshots' described above: it is positioned at a certain point in time, and it contains the amp, pitch, and pan values of every partial at that timepoint. That, of course, is a lot more information than can be represented with a single point, which is where the second display comes into play.

The larger display, in the top half of the Additive Editor view, consists of a series of **partial bars** whose levels represent the amp, pitch, pan, or phase values of every partial, depending on the mode of the display. Normally, **the values on display are those contained in the currently selected breakpoint** — so the relationship between the breakpoint envelope and the partial bars display can be depicted as follows:



In the image above, the partial bars display shows the partial amplitude values contained in the currently selected breakpoint. By changing the mode of this display, you can also view the partial pitch or pan values contained in the currently selected breakpoint (as well as the initial phase value of each partial). Finally, the partial bars display can also be switched into an OVERALL mode, in which the partial amp/pitch/pan values are are averaged across all the breakpoints. Working in OVERALL mode, you can quickly adjust the level of a certain partial across the entire sound, without having to step through individual breakpoints.

Back to the breakpoint envelopes. You now know that the timing (or *x* value) of a breakpoint represents the timing of a 'snapshot' that captures the amp/pitch/pan/phase values of every partial at that timepoint. And you know that you can inspect all of the partial values contained in a breakpoint via the linked partial bars display. But what does the level (or *y* value) of a breakpoint signify?

Normally, when the partial bars display mode is AMP, PAN, or PHASE, breakpoint levels represent partial amplitudes. The precise meaning of a breakpoint level depends on the mode to which the breakpoint envelope itself is set. In ALL PARTIALS mode, each breakpoint level represents the sum of the partial amplitudes contained in a particular snapshot. If you raise or lower a breakpoint level in ALL PARTIALS mode, you'll see all the partial amp bars rise or fall accordingly. In SELECTED PARTIAL mode, each breakpoint level represents the amplitude of the one partial currently selected in the partial bars display. (The currently selected partial is indicated with a highlight in the partial bars display; note that partial number 1 is highlighted in the image above.) If you raise or lower a breakpoint level in SELECTED PARTIAL mode, you'll see only the currently highlighted partial amp bar will respond by rising or falling.

If, instead, the partial bars display mode is PITCH, then breakpoint levels represent partial pitches. Again the precise meaning of a breakpoint level depends on the mode to which the breakpoint envelope itself is set. In ALL PARTIALS mode, each breakpoint level represents the fundamental pitch. In SELECTED PARTIAL mode, each breakpoint represents the pitch of the currently selected partial.

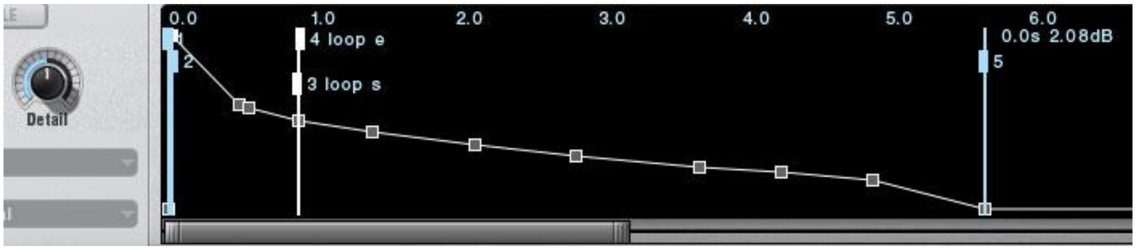
Note: If you look closely at the breakpoint envelope as you select different partials in SELECTED PARTIAL mode, you may notice (depending on the additive data) that the time positions of breakpoints vary from one partial to the next. This is one of the keys to the high quality of resynthesis in Alchemy: each partial can have an independent set of breakpoints. But this is not normally a concern for users. Whenever you work in ALL PARTIALS mode, Alchemy presents a series of breakpoints 'linked' to all of the partials, and adjustments to individual partial times and values are handled automatically behind-the-scenes.

Detail knob

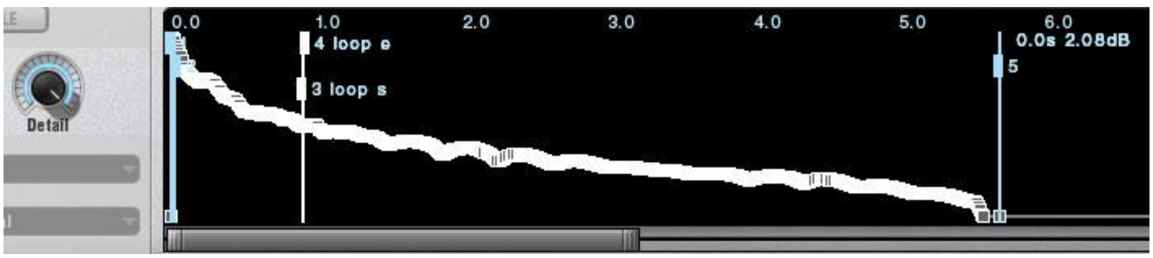
Let's import a factory sample in order to get a closer look at typical additive data. You can do this without exiting the Source Edit page, so we'll start from there.

- If you are not already viewing the Source Edit page, access it by clicking the Source A EDIT button. Then click the ADDITIVE button across the top of the Edit page to bring the Additive Editor into view.
- On the top left-hand side of this Source Edit page is a content field. (In an initialized preset, it reads 'Saw'; if you have already imported an audio file, it will show the filename; if you started with an initialized preset and have already made changes to the partial bars or breakpoint envelope, it will read 'EditorData'.) Click this field and choose Import from the popup menu.
- In the ensuing Import dialog, set the import mode to ADDITIVE and select 'FACTORY' > 'Keys' > 'Clavinet-FM' > 'Clavinet-C3.wav'. Leave the import settings at their default values and click IMPORT. (A more detailed description of this process is offered on the Import page.)

As soon as Alchemy has finished its calculations, you will be able to hear the resynthesized clavinet as you play your MIDI controller, and likewise you will be able to see the additive data displayed in the editor. We'll focus on the breakpoint envelope display at the bottom of the page. Assuming the Detail knob is at its default value of 50%, you should see a total of 12 breakpoints in the envelope — you may have to scroll or zoom the display to view them all.



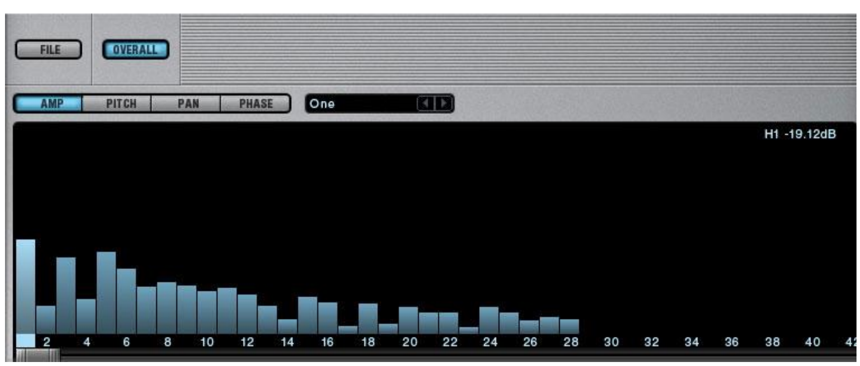
In reality, there are many more than twelve points at which partial amp, pitch, and pan values are defined. To see the complete series of breakpoints, increase Detail to 100% by turning the knob fully clockwise. You should now see something on the order of 500 breakpoints, but a precise count is beside the point. For anything other than the most excruciatingly precise sort of editing, the 100% view is more detail than you need to see.



The beauty of the Detail function is that, no matter how high or low you set it, any adjustments you make — to a breakpoint time or level, or to individual partial amp/pitch/pan/phase values in the partial bars display — are channeled through the currently selected breakpoint and applied appropriately to the underlying data.

Note: If you are programming an additive sound from scratch, rather than editing resynthesis data, you may find it useful to work at 100% Detail, so that you can see the exact data that you're creating.

Partial bars display and controls



Several controls are positioned above the partial bars display.

- The FILE button offers various commands for transferring content from breakpoint to breakpoint:
  - **Copy from Breakpoint** — Copies all the partial data contained in the currently selected breakpoint, and places it on the clipboard.
  - **Paste to Breakpoint** — Pastes all the partial data from the clipboard to the currently selected breakpoint.
  - **Paste Amp (Pitch, Pan) Data to Breakpoint** — Pastes one type of partial data from the clipboard to the currently selected breakpoint. If the partial bars display is set to AMP mode, the command is named 'Paste Amp Data to Breakpoint', and so on.
  - **Paste Amp (Pitch, Pan, Phase) Data to All Breakpoints** — Pastes one type of partial data from the clipboard to every breakpoint. If the partial bars display is set to AMP mode, the command is named 'Paste Amp Data to All Breakpoints', and so on.

A common use of 'Paste Pitch Data to All Breakpoints' is in case a resynthesized sound has unwanted pitch fluctuations. First of all, you can resolve these artifacts by reducing the PVar (pitch variation) control on the Source A/B/C/D sub-page; but this aligns each partial with its harmonic tuning, which may change the character of the sound too much. So another approach is to use 'Copy from Breakpoint' to capture data from a single breakpoint, and then to 'Paste Pitch Data to All Breakpoints'. Result: the partial pitch values of the copied breakpoint are applied to all breakpoints, so the inharmonic features of the copied breakpoint are preserved while the fluctuations from one breakpoint to the next are successfully eliminated.

- When the **OVERALL** button is off (unlit), the partial bars display shows the values contained in the selected breakpoint. When it is on (lit), the partial bars display shows values averaged across all breakpoints.
- The **AMP**, **PITCH**, **PAN**, and **PHASE** buttons determine which type of partial data is shown in the partial bars display.
- The **Edit Groups** field contains options such as 'One', 'All', 'Octaves', and so on, which are described in more detail below.

Several editing actions are performed directly in the display:

- **To adjust a partial level**, drag it up or down, or click directly at the desired height of the partial bar.

  - You can **drag left or right across multiple partials** to set their levels with a single ‘painting’ action.
  - For **fine control** over a partial level, hold shift as you drag up or down.
  - The **top right corner** of the partial bars display reports the currently selected partial number and its pitch, amp, pan, or phase value.

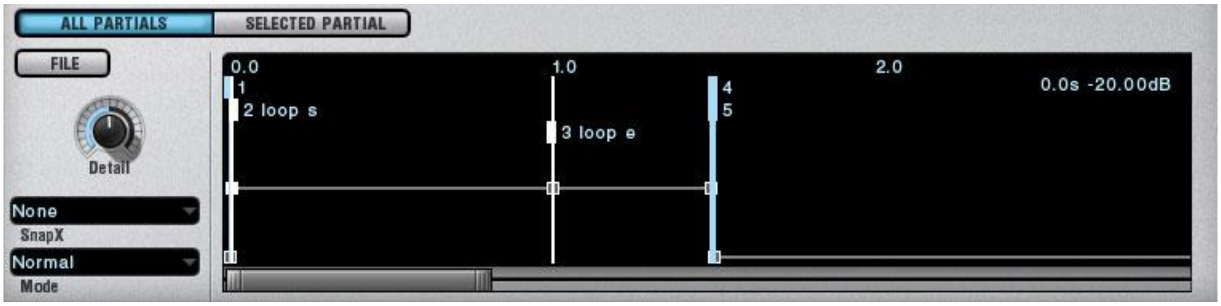
- **To set a partial level to zero**, you can drag it below the baseline; or you can right-click (control-click) it.
  - You can **set multiple partial levels to zero**, once you’ve dragged a partial below the baseline, by keeping the mouse button pressed and dragging left or right. You can also ‘zero’ multiple partials by clicking and holding the right mouse button (or control-clicking and holding a single-button mouse) and then dragging left or right.
- **To select a partial** without changing its value, click below the baseline. (Selecting a partial will bring its amplitude envelope into view when the breakpoint envelope display is in SELECTED PARTIAL mode.)
- The total number of partials may be as high as 600. To work with partials that are currently out of view, use the standard scroll bar and/or zoom handles at the bottom of the partial bars display.

Note. Higher partials that are part of the additive data may not be heard unless the NOsc (number of oscillators) control on the Source A/B/C/D sub-page is set sufficiently high. (For instance, raising the amplitude of partial number 72 will have no effect when NOsc = 60.) Conversely, setting a higher NOsc value will have no effect in ADDITIVE mode unless there is actually data for the higher partials.

- The **Edit Groups** field (above the partial bars display) lets you choose whether edits performed in the display apply to only one partial (in ‘One’ mode, which is the default) or to any of various groups (such as ‘Odd’, ‘Even’, and ‘Octaves’). For instance, if you choose the ‘Octaves’ group and then drag upwards in the display, then the fundamental and all octave-related partials (2, 4, 8, etc.) will increase.
  - If you click/drag within the range of partials for which there is existing data, then only partials within this range will respond (if they are part of the selected Edit Group). For instance, if you have analyzed a sample, and the resynthesis data uses the first 72 partials, then setting the Edit Group to ‘Odd’ and dragging upwards from a position within the first 72 partials will increase the levels of the odd partials up through 71, but it will not create new data for partials 73 or higher.
  - If you click/drag beyond the range of partials for which there is existing data, then new data will be created up to the position at which you’ve clicked/dragged. For

instance, if the existing data uses only the first 4 partials, then setting the Edit Group to ‘All’ and dragging upwards from the position of partial 12 will increase the levels of all partials up through 12, creating new data for partials 5–12.

Breakpoint envelope display and controls



The area above and to the left of the breakpoint envelope display is home to a number of controls.

- The **FILE** button gives access to a pop-up menu in which you can **Clear** all data from the current Source, or translate a picture into additive data via the **Import Image** command, which is described in more detail towards the bottom of this page.
- A pair of **display mode** buttons determines how information is presented in the breakpoint envelope display. In **ALL PARTIALS** mode, the breakpoint levels represent the sum of all the contributing partial amplitudes (or the rise and fall of the fundamental pitch). In **SELECTED PARTIAL** mode, the breakpoint levels represent amplitude values of the one partial that is selected (and highlighted) in the partial bars display (or the pitch fluctuations of the currently selected partial).
- The **Detail** knob controls the resolution of the breakpoint envelope display. It is described in more detail in a separate section, above.
- The **SnapX** field allows you to constrain the time position when you move or create breakpoints. The ‘None’ setting leaves these actions unconstrained, while the ‘4’/‘8’/‘16’/‘32’/‘64’ settings constrain them to increasingly fine grids. (At the highest SnapX settings, you’ll need to zoom in to see all the gridlines.)
- The **Mode** field controls the behavior of the envelope when a breakpoint is dragged. In ‘Normal’ mode, other points remain stationary when one point is dragged. In ‘Slide’ mode, dragging one point causes all the subsequent points the move in tandem, so that the relative distance between these points is preserved. In ‘Stretch’ mode, dragging left compresses earlier points and stretches later points, while dragging right stretches earlier points and compresses later points; in either case the total length of the envelope is preserved.

> ***Notice that breakpoint levels remain fixed in 'Stretch' mode, so you can only drag left/right.***

Working in the breakpoint envelope display is much like working with any of Alchemy's multi-segment envelopes (see the MSEG page for details). Here are some additional things you should know:

- The time and level of the selected breakpoint are reported in the parameter value display at the top of the interface.
- When the Additive Editor is in PITCH mode, breakpoint levels represent *partial pitches*. (You can edit these levels to eliminate a pitch wobble, or to create vibrato.) When the Additive Editor is in any other mode, breakpoint levels represent *partial amplitudes*.
- Unlike MSEGs and ModMaps, the additive breakpoint envelope always consists of straight-line segments with no convex/concave curvature.
- The very first and very last breakpoints always have amplitudes of zero; in sounds with fast attacks, you'll find a second breakpoint with a nonzero amplitude at a time position very close to the first breakpoint. (You may need to increase the Detail setting in order to find that second breakpoint.)
- In the display, the warp markers are superimposed on the breakpoint envelope. For a full explanation of the warp markers, see the main Source Edit page.
- Most of the things you can do with warp markers on the Source Edit page can be done here in the breakpoint envelope display as well.
    - You can double-click away from existing warp markers to create a new one, and double-click on an existing warp marker to delete it.
    - You can right-click a warp marker to assign it the role of loop start/end.
- When you drag warp markers in the breakpoint envelope display, they always snap to existing breakpoints. (Compare the main Source Edit page, where warp markers can be dragged freely to any position.)
- Positioning the loop start and end markers both on the same breakpoint creates a sustain point, which means you can listen to that point in the sound for as long as you wish. By doing this repeatedly at different breakpoints, you can 'investigate' successive snapshots of the additive data in order to locate ones you may wish to edit.

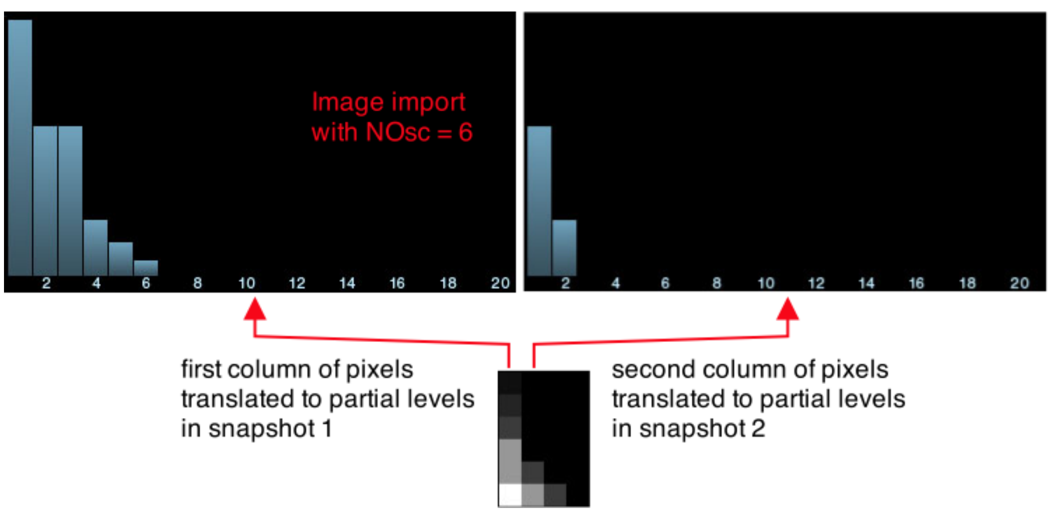
Image import

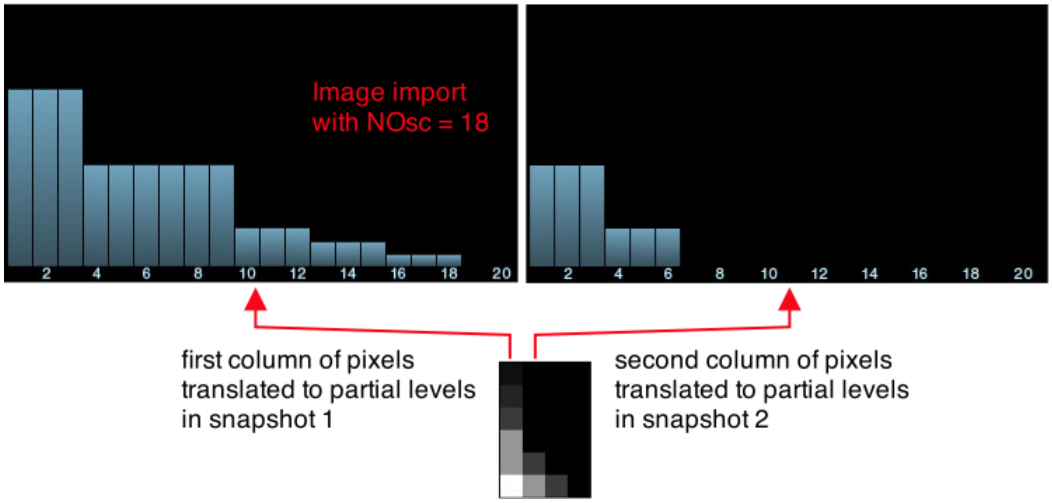
Alchemy is capable of importing images (in PNG, or Portable Network Graphics, format) and translating them into additive synthesis data. You can easily convert existing images into sounds, and you can design new sounds by using (external) graphics software and drawing the image you want to import.

> ***Note that Alchemy's Spectral Editor has its own image import capabilities, and you can get characteristically different results by importing an image into either the additive or the spectral engine.***

The translation from image to additive data works as follows:

- **Each column of pixels represents a 'snapshot' in the additive data.** The leftmost column describes snapshot 1, the next column to the right describes snapshot 2, and so on. (These snapshots are timed at a steady rate of 20 per second.)
- Within each column, the pixels from low to high translate to partials from 1 up to the limit set by NOsc. Thus you can use the NOsc control (on each Source sub-page) to 'scale' the vertical dimension. If your image is 100 pixels tall, setting NOsc = 100 creates a one-to-one correspondence between pixels and rows of partials: the bottom row represents, partial 1, the second row up represents partial 2, and so on. Setting NOsc less than 100 will compress the image into a smaller range of partials, while setting NOsc greater than 100 will stretch the image across a wider range of partials.
- **The brightness of each pixel determines the amplitude** of a particular partial in a particular snapshot. A black pixel corresponds to silence, while a white pixel corresponds to maximum amplitude. (You can import both color and grayscale images; color information other than brightness is discarded.)





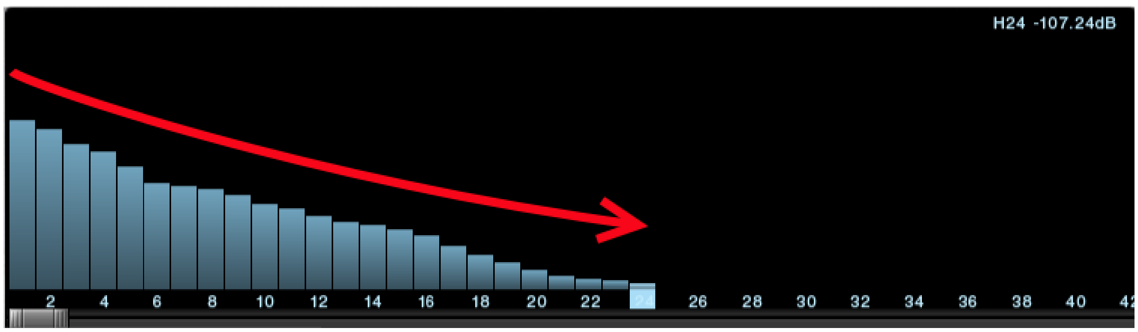
Importing an image with many bright pixels gives you additive data with many high-amplitude partials, which may cause clipping to occur. If the results of an image import are unexpectedly noisy, you can reduce the overall amplitude in a number of ways. First, you can reduce the Vol setting of the additive element. Second, you can edit the data by setting the Additive Editor to OVERALL mode, selecting the 'All' edit group, and dragging downwards in the partial bars display. And third, you can use your graphics software to darken the image before importing it.

Example: Creating a simple additive sound from scratch

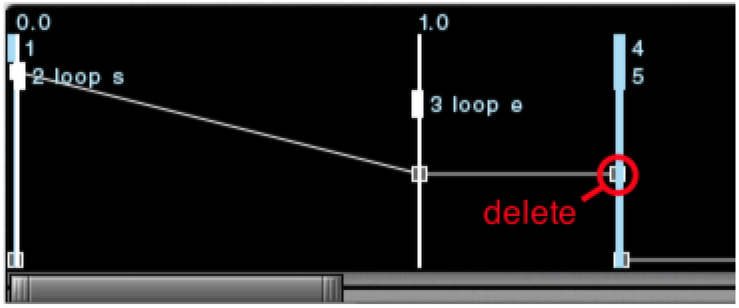
In this example, we'll introduce some basic principles of additive programming in Alchemy by creating a simple 'plucked' sound from scratch.

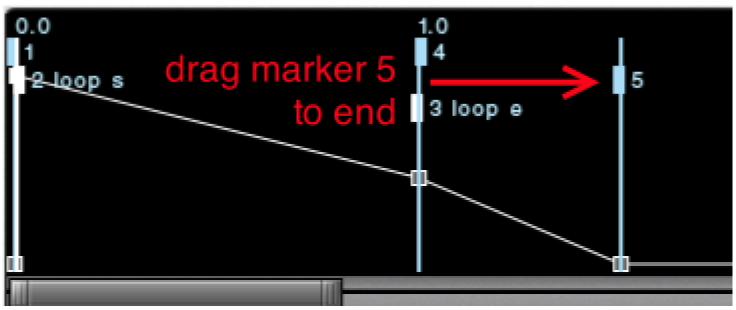
- **Initialize Alchemy** by choosing the 'Clear' command in the Title bar's FILE menu.
- Go to the Source A sub-page, just to get a better look at the **current state of the preset**. When initialized, Source A is in VA mode, the oscillator waveform is set to the basic 'Saw', NOsc = 1, and the Pitch profile is set to 'Unison'. For additive synthesis, we normally want an entirely different configuration. We need Source A to be in ADD mode, and we typically want the oscillator waveform set to the basic 'Sine', with a larger value of NOsc (permitting many partials to be active), and with the Pitch profile set to 'Harmonic'. However, it is not necessary to change any of these settings from their initial values — **as soon as you enter the Additive Editor and start creating data, the configuration of Source A switches, automatically, to a sensible set of defaults for additive programming**. So let's get right to work…

- Click the Source A EDIT button to open the Source Edit page, and then click ADDITIVE at the top of the page to bring the Additive Editor into view.
- Since we're starting from scratch rather than editing resynthesized data, let's increase Detail to 100% for an accurate view of all the data. (The breakpoint envelope display won't appear any different, since there are not yet any additional details to be shown.)
- If the second breakpoint in the envelope is not already selected, click it. (This is a typical starting point, since the first breakpoint always sits at time zero and level zero, and can't be altered.)
- Now ensure that the partial bars display is set to AMP mode, and that OVERALL mode is off, and draw in some bars to define the harmonic content of the beginning of the sound. Since our goal is a 'plucked' sound, we want a bright sound with a significant number of active partials at the beginning, which will then decay to a darker sound further on. A design such as the following one, which can be created in a single sweep across the display, works well. But you needn't copy it precisely.

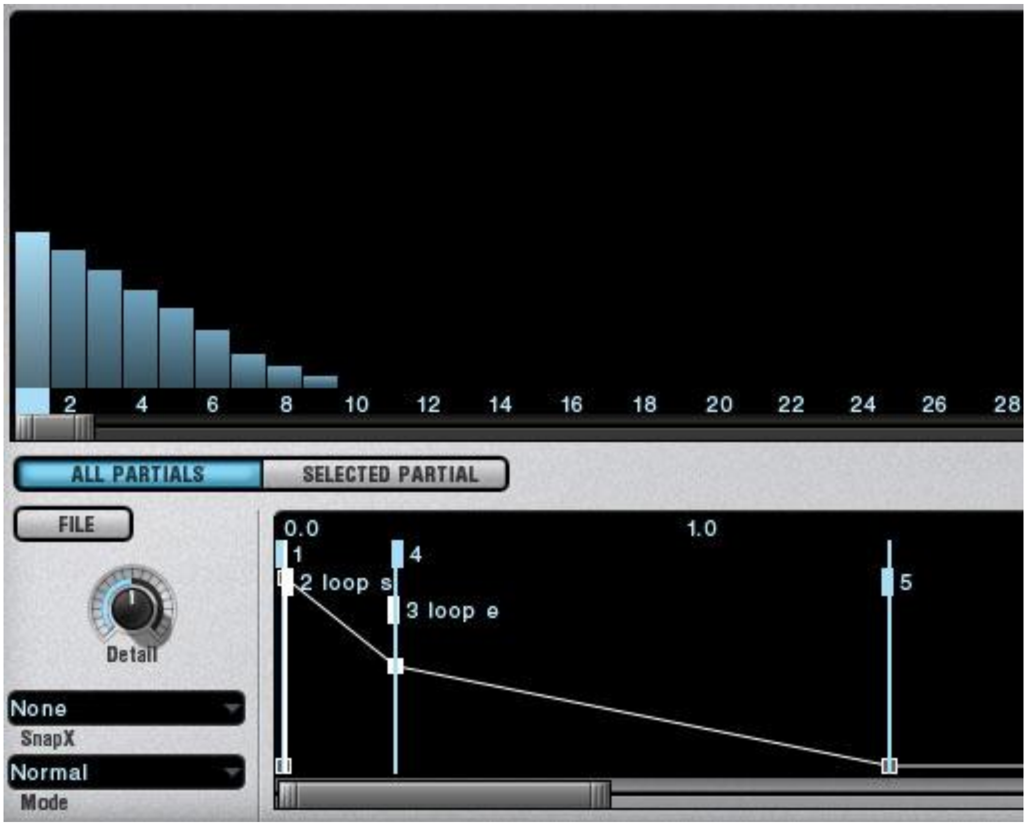


- Play a few notes on your MIDI controller to confirm that the sound now begins with a bright timbre as intended. This is a step in the right direction, but more needs to be done. For instance, the default loop mode is 'Continuous', so the sound will loop indefinitely when you hold a note. To switch looping off, change the Loop mode to 'None' in the field at the far left of the Edit page.
- Progress. But now the end of the sound falls off too abruptly instead of decaying smoothly. A quick solution is to delete the second-to-last point. (Afterwards, drag warp marker 5 back to the final point.)





- Now the results sound reasonably good. But let's re-shape the decay so that most of the energy falls off quickly and the tail of the decay is less dramatic. Select breakpoint 2 again, and then choose 'Copy Breakpoint' from the partial bars display FILE pop-up menu. Then select breakpoint 3, drag it most of the way towards breakpoint 2, and choose 'Paste Breakpoint' from the partial bars display FILE pop-up menu. Then reduce or eliminate the higher partials in the data you've just pasted. With breakpoint 3 still selected, the final result should look something like this:


2
4
6
8
10
12
14
16
18
20
22
24
26
28
ALL PARTIALS
SELECTED PARTIAL
FILE
Detail
None
SnapX
Normal
Mode
0.0
1.0
1
4
2 loop s
3 loop e
5

Spectral Editor

Once you have accessed the Source Edit page, clicking the SPECTRAL button at the top of the right-hand half of the page will bring Alchemy's Spectral Editor into view.

Overview

The Spectral Editor serves two purposes. First, it allows graphical editing of spectral resynthesis data. And second, it allows you to design sounds from the ground up by working with simple 'paint' tools directly on the 'canvas' of the the graphical display.

Spectral data is displayed in the editor according to the following principles:

- **Time** (in sec) is represented along the *x* axis (from left to right).
- **Frequency** (in Hz) is represented along the *y* axis (from bottom to top).
- **Amplitude** is represented by brightness, using shades of blue. The full range, from silent (black) through loudest (white) looks like this:

Note that accurate resynthesis requires a much finer frequency resolution than the spectral display can accommodate. Therefore the frequency information depicted in the display is a somewhat coarse representation of the underlying data. Editing and creating data graphically is performed at the resolution of the display. This means, for instance, that you cannot paint conventional melodies and chords consisting of precise notes.

Spectral editor controls

- Click the **FILE** button to access a pop-up menu with the following commands:

- **Import Image** — Opens a dialog in which you can select a file in PNG fomat to be placed as an image on the spectral canvas. The imported image is placed at the far left of the canvas. The height of the image is scaled to fit the entire vertical range of the canvas, and the width of the image is scaled by the same factor as the height (so the proportions of the original image remain intact). The imported image's color information is discarded, and its brightness information is mapped to the shades-of-blue amplitude scale described above. If the newly placed image does not extend fully to the right-hand edge of the canvas, then existing data that lies beyond the right edge of the newly placed image remains in place.

- **Import Image to Brush** — Opens a dialog in which can select a file in PNG format to be used as a brush for painting on Alchemy's spectral canvas. Unlike importing an image directly onto the canvas, importing it as a brush preserves the original dimensions of the image and gives you full control over the position at which the image is placed. (As with any brush, you can 'print' an image of the imported brush by clicking or paint strokes with it by dragging.)

- **Undo** — Reverses the last change made to the spectral data.

- **Clear** — Deletes all spectral data from the source, leaving only silence (solid black image).

* **SELECT** and **BRUSH** buttons — When the editor is in SELECT mode, clicking and dragging across the spectral canvas defines a rectangular selection. If you then switch to BRUSH mode, the selection you've just defined can immediately be used as a brush shape, which remains available until you select a different brush shape.

* The **Shape** field offers a choice of several pre-defined brush shapes, each with a descriptive name.

* The **Colour** knob adjusts the amplitude scaling of the brush. White brush pixels paint white on the canvas when Colour is 100%, medium blue on the canvas when Colour is 50%, and black on the canvas when Colour is 0%. Brush pixels representing lower amplitude values are scaled accordingly; for example, medium blue brush pixels paint medium blue on the canvas when Colour is 100%, dark blue on the canvas when Colour is 50%, and black on the canvas when Colour is 0%.

* The **Opacity** knob allows for transparency effects. When Opacity is 100%, the paint modes behave as described below (see 'Paint mode buttons'). When Opacity is 50%, the pixel values that result from painting in any mode are halfway between the pixel values in the existing image and the pixel values that would result from painting with full Opacity. (Note that an image imported directly to the canvas will always have full Opacity; on the other hand an image imported as a brush can be used to paint with any degree of Opacity.)

* The **Paint mode** buttons determine how the brush interacts with the existing canvas image.

- In **SET** mode, each pixel on the canvas is set identically to the corresponding brush pixel. In this mode, painting with black replaces the existing image with silence.
- In **ADD** mode, the brush pixel values are added to the existing canvas pixel values. In this mode, painting with black leaves the existing image unchanged, while painting ‘multiple coats’ of blue results in higher and higher amplitudes the more coats you paint.
- In **MUL** mode, the brush pixel values are multipled with the existing canvas pixel values. In this mode, painting *with* black on an image of any color, and painting *on* black with paint of any color, both result in black. You can use a black brush in MUL mode as an eraser, and you can use a brightly-colored brush in MUL mode to boost the portions of an image that already produce sound, without disturbing the portions that are silent.
- Finally, in **DEL** mode, the brush pixel values are subtracted from the existing canvas pixel values. In this mode, a bright white brush works like an eraser.

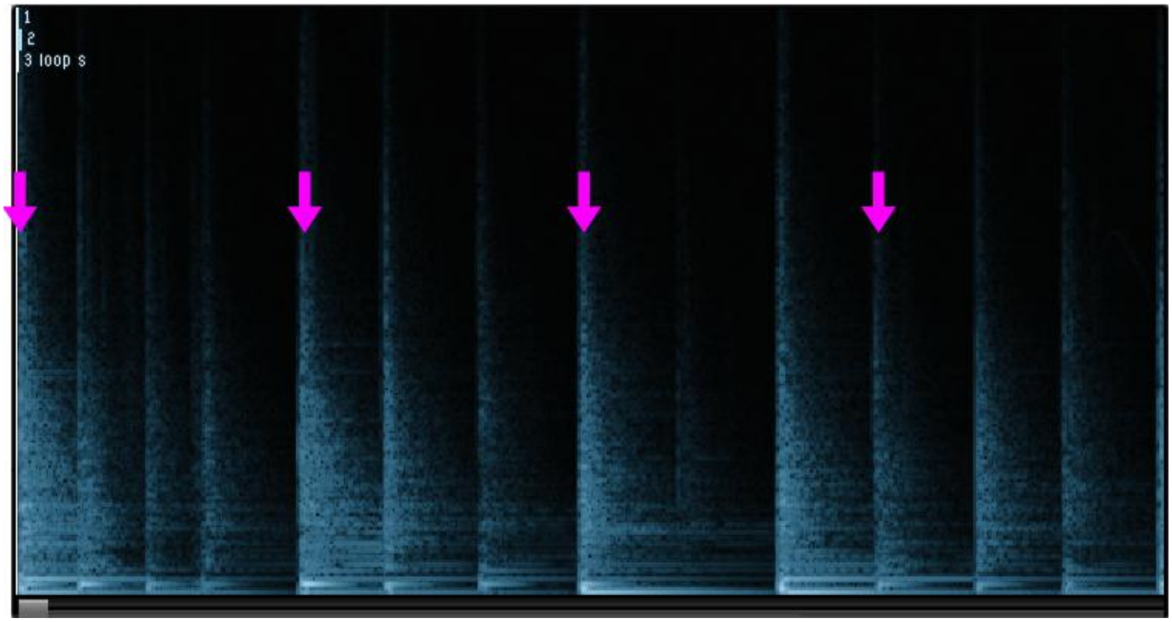
Spectral canvas

Once you are familiar with the brush modes, the relation between colour and amplitude, and the distribution of frequencies from the bottom to the top of the canvas, working in the spectral editor’s graphical display is an intuitive process. Here are some useful techniques and related information.

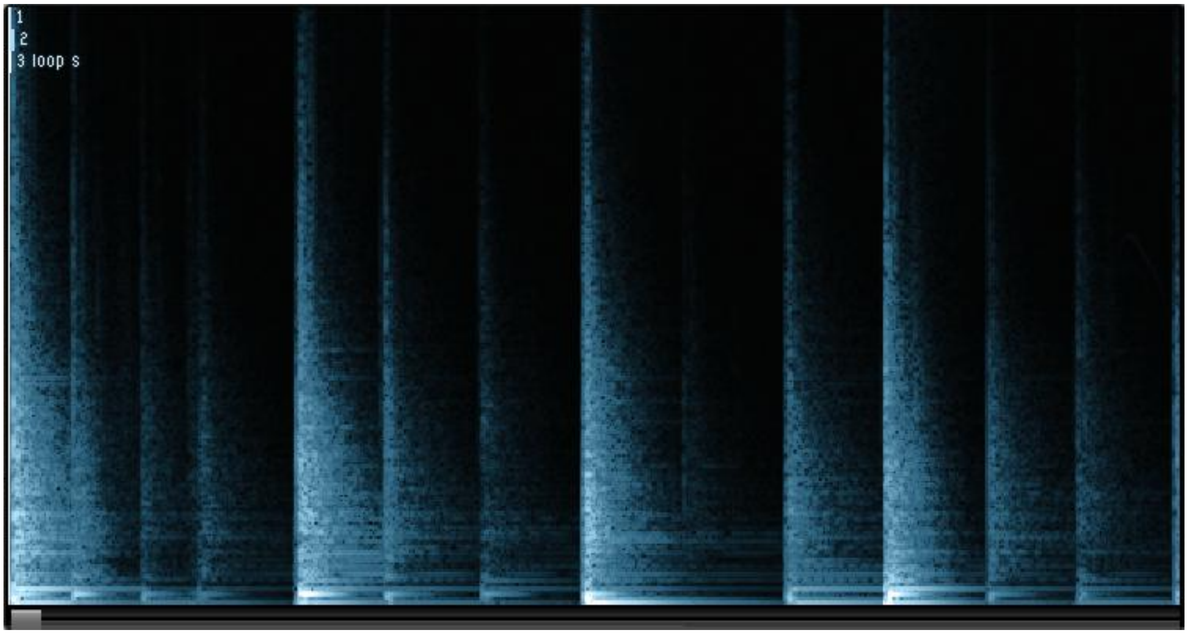
- The time (in sec) and the frequency (in Hz) of the brush (or selection cursor) position are reported in Alchemy’s parameter value display. If data is present at, say, the 2500 Hz position, this means that the output of the spectral element will include a 2500 Hz component when C3 is played (or when keytracking is off and any note is played).
- To constrain painting vertically, hold Shift while painting.
- To constrain painting horizontally, hold Control (Windows) or Command (Mac) while painting.
- Warp markers are superimposed on the canvas and can be moved freely by dragging their handles. To create or delete warp markers, or re-assign the loop start and end markers, switch to the main Source edit view by clicking the MAIN button at the top of the editor.

Example: Modifying a drum loop

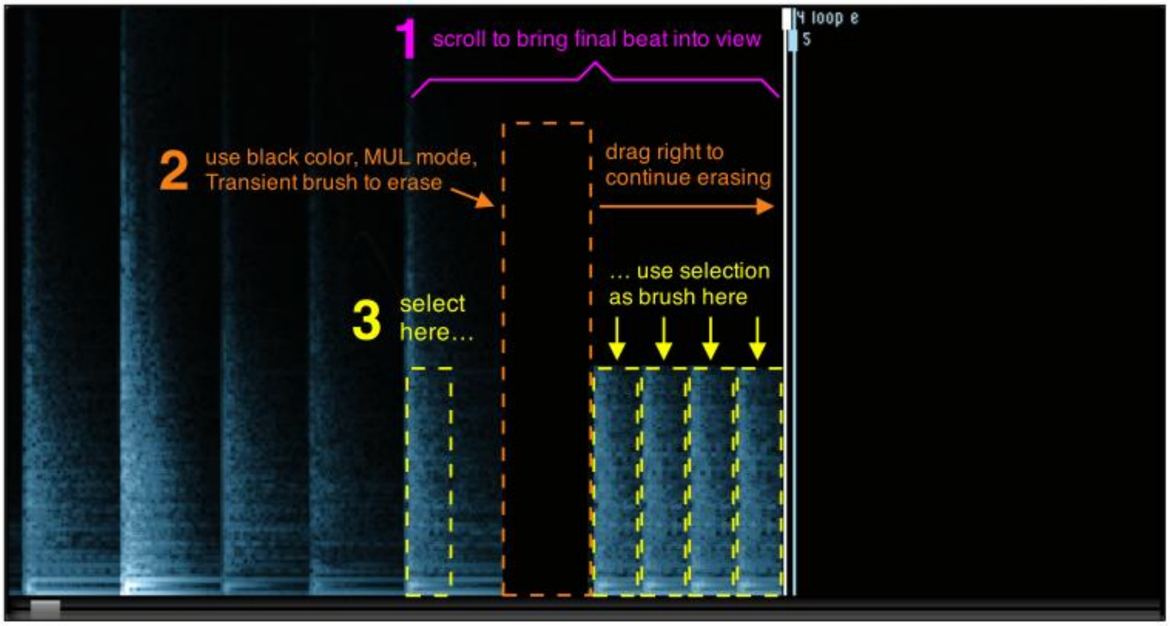
- **Initialize Alchemy** by choosing the ‘Clear’ command in the Title bar’s FILE menu.
- **Import** (into Source A) the factory sample ‘Loops’ > ‘PercussiveLoops’ > ‘DjembeLoop2-4bts’. Be sure to set the import mode to SPECTRAL. (See the Import page for more details of this process.)
- The imported image should look as follows. In this illustration, a superimposed series of arrows marks certain drum hits that we will now modify.



- Select the brush shape ‘Transient’, which is designed for creating or boosting attack transients. We’ll use this tool to add ‘punch’ to the four drum hits marked above. Set the colour to light blue (Colour knob turned to approximately 75%), and set the brush mode to MUL. Carefully align the brush so that its bright left edge coincides with the peaks marked above, and click one or more times in each of these positions, until the result looks something like the following. (Play and hold C3 on your MIDI controller to hear the effect of these edits.)



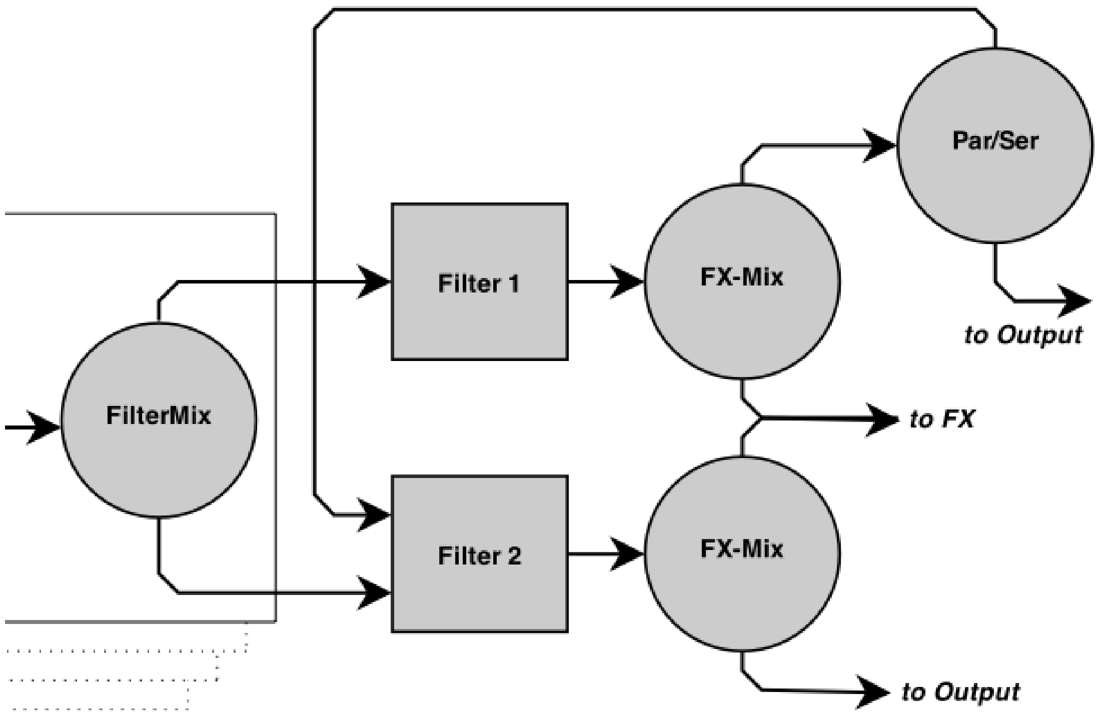
- Let's make some additional changes to the end of the loop.
    - Scroll the spectral canvas to the right until the final beat is in view. (All of the actions described below are summarized in the illustration that follows.)
    - With the 'Transient' brush still selected, and the brush mode still set to MUL, reduce the Colour value all the way to black (Colour knob turned fully left). Then align the brush with the position one sixteenth note after the final beat; click at this position, and drag to the right to erase all the spectral data from this position to the end. (Play and hold C3 again on your MIDI controller to confirm that this produces a silent region at the end of the loop.)
    - Next, switch to the SELECT tool and select the first half of a medium-loud drum hit. Switch back to the brush tool, turn the Colour knob fully right, and set the brush mode to SET. You can now paint using the selected image as a brush. Click four adjacent positions at the end of the loop to produce a 'roll' effect. (Play and hold C3 once more on your MIDI controller to hear the final result.)
- These last three steps are summarized, alongside the results they produce, in the following illustration.



1 scroll to bring final beat into view
4 loop e
5
2 use black color, MUL mode, Transient brush to erase
drag right to continue erasing
... use selection as brush here
3 select here...

Filter

There are two Main filter modules, labelled **Filter 1** and **Filter 2**.



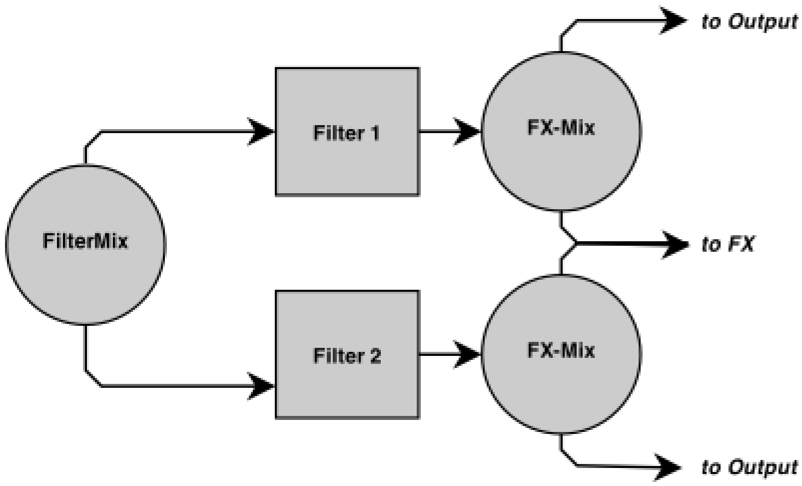
The outputs from all four Sources are sent (in varying amounts, according to their respective FilterMix settings) to the Main filters.

Both Main filter modules provide multi-mode filters with identical controls. Each one has its own **FX-Mix** control, which (like the FilterMix control in a Source) works as a kind of two-channel mixer, or crossfader.

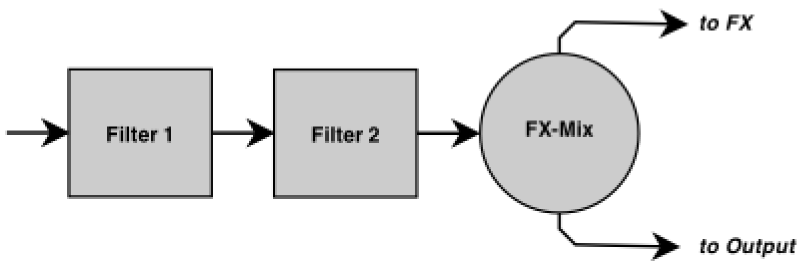
- With the FX-Mix control turned all the way to the left, all of the filter's output is sent to Alchemy's main output, and none of it to the Effects module.
- With the FX-Mix control turned all the way to the right, all of the filter's output is sent to the Effects module, and none of it to Alchemy's main output.
- With the FX-Mix control set half-way, equal amounts of signal are sent to the Effects module and Alchemy's main output.

Finally, the **Par/Ser** knob acts as one more crossfader, allowing you to send the non-FX portion of the Filter 1 signal to the Output stage (when Par/Ser = 0%), to the input of Filter 2 (when Par/Ser = 100%), or to a mix of these destinations (when Par/Ser has some intermediate value). Therefore:

- ***To use the Main filters in parallel*** — set the Source FilterMix controls as desired, set Par/Ser to 0%, and set the Filter 1 FX-Mix control as desired.



- ***To use the Main filters in series*** — set the Source FilterMix controls to '100% F1 0% F2', set the Par/Ser knob to 100%, and set the Filter 1 FX-Mix control to 0%.



Note that when the Par/Ser knob is set to 100%, a portion of the Filter 1 signal will bypass Filter 2 whenever the Filter 1 FX-Mix control is set above 0%.

Global page controls

The Filter section on the Global Page provides access to the two Main filter modules, each of which offers a powerful multi-mode filter.

Identical controls are provided for Filter 1 and Filter 2:

- **On** toggles the filter unit on or off.
- The **filter type** selection field allows you to set each filter to one of 50 filter types (see 'Filter Types' below).
- The three ***primary filter controls*** are **Cutoff**, **Res**, and **Drive**. The meaning of these controls depends on the selected filter type (see 'Filter Types' below).
- **FX-Mix** controls how much of the filter module's output is sent to Alchemy's Effects section, and how much directly to the main Output.

Filters and filter types

Altogether, Alchemy provides a total of 15 multi-mode filter modules (two Main filters, four sets of three Source filters, and the MMFilter module in the Effects section). There are 50 filter types to choose from in the Main filters and the MMFilter module. The Source filters offer a smaller selection of filter types, and they lack the Drive control found in the Main filters and the MMFilter module.

You will find descriptions of all 50 filter types below.

Filter types: lowpass, bandpass, hipass

These filter types will be familiar to many users. A **lowpass (LP)** filter passes the portion of a sound below a specified cutoff frequency and rolls off the portion above that frequency. A **bandpass (BP)** filter passes the portion of a sound occupying a band surrounding the cutoff frequency and rolls off the portions above and below that band. And a **highpass (HP)** filter passes the portion of a sound above a specified cutoff frequency and rolls off the portion below that frequency.

There are 42 different LP, BP, and HP filter designs available in Alchemy, each with distinctive characteristics that you may prefer for a given purpose. The three **principal filter controls** have standard functions for all 42 of these filters:

- **Cutoff** controls the filter cutoff frequency.
- **Resonance** controls the filter resonance or emphasis, such that higher settings boost frequencies in the immediate vicinity of the cutoff frequency.
- **Drive** allows the filter to be overdriven; the precise effect varies with each filter design.

The available LP, BP, and HP filter designs include:

- **LP2-BQ**, **BP2-BQ**, **HP2-BQ** — 2-pole, bi-quad filters.
- **LP2-Fat**, **BP2-Fat**, **HP2-Fat** — 2-pole filters designed to saturate heavily at higher Resonance and Drive settings.
- **LP2-SVF**, **BP2-SVF**, **HP2-SVF** — 2-pole state-variable filters.
- **Analog-modelled filters** — there are several designs modelled on the filters in classic hardware synths. Each of these is available in **two versions**, a normal one and a 2x-oversampled **HQ** one (offering higher quality at the cost of greater CPU usage).
    - 2-pole analog-modelled filters — **LP2-K20**, **BP2-K20**, **HP2-K20**; **LP2-XP**, **BP2-XP**, **HP2-XP**; **LP2-MG**, **BP2-MG**, **HP2-MG**.
    - 4-pole analog-modelled filters (these provide a steeper roll-off of frequencies beyond the cutoff) — **LP4-XP**, **BP4-XP**, **HP4-XP**; **LP4-MG**, **BP4-MG**, **HP4-MG**.

Filter types: formant and peaking

The **formant filter** works like a bandpass filter with a potentially very narrow bandwidth. It is designed to mimic a formant, or fixed resonance, such as those produced by the hollow body of a guitar or by the shape of the vocal cavity (which is adjusted during speech to produce different vowel sounds).

- **Cutoff** controls the resonant frequency of the formant.
- **Resonance** controls the width of the band surrounding the resonant frequency. Low values produce a narrow band, high values a wider one. Setting the resonance too low may cause little or no sound to pass through the filter.
- **Drive** has no effect for this filter type.

Note: To create multiple formants, you can configure the Source Filters in parallel and set the type of all three filters to 'Formant'. See the example at the bottom of this page, which uses multiple formants to mimic vowel sounds.

The **peaking filter** boosts a narrow band around some resonant frequency while leaving the rest of the signal more or less unchanged.

- **Cutoff** controls the resonant frequency of the boosted band.
- **Resonance** controls the amount of boost. Higher values are generally the most effective.

- **Drive** has little effect for this filter type beyond boosting the overall gain.

Filter types: comb filters

A **comb filter** mixes the original signal with one or more copies delayed by a very short time interval. At some frequencies this mixture causes phase cancellations, while at other frequencies it causes reinforcements; the result is a ‘spiky’ frequency response with multiple resonant peaks. (A graph of these peaks resembles the teeth of a comb, which gives this filter type its name.)

Alchemy offers two comb filter designs. **CombP** (P is for ‘plus’) uses positive feedback on the delay lines, while **CombM** (M is for ‘minus’) uses negative feedback to produce less extreme effects, often with a ‘hollow’ quality.

- **Cutoff** controls the delay time in the comb circuit (lower cutoff = longer delay).
- **Resonance** controls the amount of feedback in the comb circuit.
- **Drive** has little effect for the comb filter types beyond boosting the overall gain.

Note that sending a percussive sound into a highly resonant comb filter will cause it to ‘ring’ at a frequency determined by the delay time (hence controlled by the Cutoff knob).

Filter types: ring modulation

Ring modulation is a process in which a modulator and a carrier signal are multiplied. Each frequency component of the modulator interacts with each frequency component of the carrier to produce two sidebands: a sum and a difference (carrier – modulator). When an Alchemy filter is set to the **RingMod** type, the signal entering the filter acts as the modulator, while the carrier is supplied internally by the filter.

- **Cutoff** controls the carrier frequency.
- **Resonance** applies a constant offset to the carrier.
    - At **0% Resonance**, the carrier wave varies between –1 and +1, and the result is classic **ring modulation**.
    - At **100% Resonance**, the carrier wave varies between 0 and 1, and the result is classic **amplitude modulation**. In this case, the carrier signal itself is present alongside the sum and difference sidebands.

- **Drive** controls the carrier waveform.
  - **Setting Drive to 0%** gives a ***pure sinewave*** carrier. This setting is often the most useful for producing characteristic 'bell-like' timbres.
  - **Setting Drive above 0%** progressively ***truncates*** the carrier waveform (reducing it to just the first quarter of the sine shape when Drive is 100%). These settings produce a greater density of sidebands, typically resulting in noisy/grungy timbres.

Filter types: distortion

The Alchemy filters offer several distortion effects.

- **Tube** — implements the the well-known and well-loved 'Tube' distortion effect from CamelPhat.
- **Mech** — A more 'metallic' sounding distortion, also from CamelPhat.
- **BitRed** — A harsher sounding 'bit reduction' effect (also known as 'bit crushing').

> ***Note that distortion effects created in the Source Filter and Main Filter stages of the signal path are 'polyphonic' — that is, each voice is distorted independently, so there are no intermodulation effects when you play chords with this kind of distortion. In contrast, the Distortion module in the global Effects stage at the end of the signal path processes a mix of all the voices.***

The principal filter controls work as follows when the filter type is set to a distortion effect:

- **Cutoff** controls the intensity of the distortion effect.
- **Resonance** controls the mix between clean and distorted signals (0% = clean only, 50% = equal mix, 100% = distorted only).
- **Drive** has little effect for the distortion types beyond boosting the overall gain.

Example: Creating vowel sounds with parallel source filters

Vowel sounds in human speech are the result of formants produced by changing the shape of the vocal cavity. By mimicking these formants, Alchemy can produce speech- or song-like effects, even in VA mode (i.e. without relying on samples of actual speech). Here is a basic example.

- **Initialize Alchemy** by choosing the 'Clear' command in the Title bar's FILE menu.
- Go to the Source A sub-page and adjust the following settings:
  - Increase **Amp** to 0 dB.
  - Set **Coarse Tune** to –12 semis.
- Click the Source A **Fine Tune** knob to bring its mod rack into view in Alchemy's Mod section.
  - In the first mod-rack slot, choose 'LFO' > 'LFO 1', and adjust the mod depth to approximately 50%.
  - In the LFO 1 control panel, set **Attack** to approximately 0.50 sec, turn off **SYNC**, and set **Rate** to approximately 5 Hz.
- Turn on the ***Source A Filter section*****,** and click the Filter button to bring the Source Filter details into view.
  - Turn on all three of the ***individual Source Filters***, and set the configuration to **parallel**. We'll come back to these filters and adjust the Cutoff and Resonance settings in a later step.
- Switch the Source section view to ALL. Click in the ***Source A content field*** and choose 'Copy Source'. Then click in the ***Source B content field*** and choose 'Paste Source', click in the ***Source C content field*** and choose 'Paste Source' again, and click in the ***Source D content field*** and choose 'Paste Source' a final time.
- On each of the Source Filter sub-pages, adjust the following settings (all values are approximate):
  - Source A — 'Ah' as in *father*
    - Cutoff 1 = 800 Hz, Res 1 = 100%
    - Cutoff 2 = 1200 Hz, Res 2 = 100%
    - Cutoff 3 = 2800 Hz, Res 3 = 100%
  - Source B — 'Ee' as in *peace*
    - Cutoff 1 = 230 Hz, Res 1 = 100%
    - Cutoff 2 = 2600 Hz, Res 2 = 75%
    - Cutoff 3 = 3200 Hz, Res 3 = 75%
  - Source C — 'Oo' as in *food*

        - Cutoff 1 = 200 Hz, Res 1 = 100%
        - Cutoff 2 = 880 Hz, Res 2 = 65%
        - Cutoff 3 = 2400 Hz, Res 3 = 50%
    - Source D — ‘Eh’ (as in *let*)
        - Cutoff 1 = 530 Hz, Res 1 = 65%
        - Cutoff 2 = 1850Hz, Res 2 = 75%
        - Cutoff 3 = 2500Hz, Res 3 = 50%
- Now that each Source is configured to produce a vowel sound, let’s set up morphing between these sounds and control it via the modwheel.
    - Switch the Source section view once more to ALL. Set the MORPH mode to ‘morph linear’ and set the X knob (the only knob) to 0%.
    - The Morph X knob’s mod rack should now be in view in the Mod section. Since the modwheel is linked by default to Performance Control 7, click in the first mod-rack slot and choose ‘Perform’ > ‘Control7’. Leave the mod depth at its default value of 100%.
    - Now as you play notes, you can use the modwheel to morph between vowel sounds.
- If the output level seems low, you can boost it by increasing the Master Amp and/or Volume knobs. (And a further boost, should you need one, is available if you load a Compressor module in the effects rack and adjust the Amount knob.)

Master

Once the signal path of an individual voice in Alchemy has left the Source section and passed through the Main filters, it reaches the **Master** section, where controls are provided for the Amplitude, Pan, and Coarse and Fine tuning of the whole voice before it mixes with other voices at the input of the Effects stage.

The Master section also provides high-level control over various attributes of a preset such as polyphony and micro-tuning. The available controls are described below.

Master voice controls

- **Amp** — Adjusts the level of the voice. By default, AHDSR 1 modulates this parameter, thereby acting as the master amplitude envelope.

  Note: Whenever an AHDSR is assigned to modulate Master Amp, the modulation depth is 'locked' and can't be adjusted. This is because setting the depth below its default of 100% would cause notes to remain above zero amplitude indefinitely — so it protects against 'stuck' notes. (If you replace the AHDSR with another type of modulator, such as an MSEG, you'll want to preserve a mod depth setting of 100% unless you really intend for notes to sustain indefinitely.)

- **Pan** — Adjusts the stereo position of the voice. (Acts as a 'pan' control for mono sounds and as a 'balance' control for stereo sounds.)

- **Coarse** and **Fine Tune** — Adjusts the pitch of the voice in semitones (Coarse) and cents (Fine).

Other Master controls

- Trigger **Mode** field — determines the conditions under which (1) a trigger signal is generated and (2) a portamento glide occurs. (Interacts with the Voice and Porto settings, described below.)
  - **Always**. ***If Voices = 1***, then a trigger is generated at the start of each legato group, and portamento occurs at the start of every note. ***If Voices > 1***, then a trigger is generated at the start of every note, and portamento occurs at the start of every note.
  - **Retrigger**. Regardless of the Voices setting, a trigger is generated at the start of every note, and portamento occurs at the start of every note.
  - **Legato**. ***If Voices = 1***, then a trigger is generated at the start of each legato group, and portamento occurs at the start of each legato group. ***If Voices > 1***, then the behavior is like the one-voice case when you play single notes in succession; but when you play a chord, each note of the chord gets its own trigger. (To count as a chord, each note needs to be within 200 msec of its predecessor.)

  NOTE. A 'trigger' is a signal that causes certain processes to execute from the beginning. These processes include the playback of audio data (such as a sample or additive/spectral data) and various modulator types, including LFO, AHDSR, MSEG, and Sequencer. In addition to the master Trigger mode described above, individual modulators have their own TRIGGER buttons. Turning a TRIGGER button off causes the individual modulator to trigger in 'Legato' mode (or not at all, if it is an LFO), regardless of the Master Trigger mode.

- **Voice** count field — Determines the maximum polyphony of a preset (up to 32 voices). Setting Voice = 1 makes a preset monophonic.
- ***Portamento controls***. The portamento mode can be set to **Rate** or **Time**. 'Rate' means that Alchemy glides from note to note at a fixed rate set by the **Porto** knob (so gliding a greater distance requires more time). 'Time' means that Alchemy glides from note to note during a fixed amount of time set by the **Porto** knob (so gliding a greater distance occurs at a faster rate).
- **Pitchbend Up** and **Down** —Determines the response to upwards and downwards pitchbend messages. Normally, you should set a positive 'Up' value (the default is +2) and a negative 'Down' value (the default is –2).

  Note that individual Sources can be set to respond to pitchbend or ignore it via the Keytrack field on each Source sub-page. Pitchbend is also available as a modulator (in the 'Note Property' category), giving

you more individualized control over the pitchbend response of each Source (as well as the ability to route pitchbend messages to mod targets other than pitch).

- **Tuning** field — Offers a vast selection of alternative tunings. Each tuning is defined by a **.tun** file stored in \Libraries\Tuning (PC) or /Libraries/Tuning (Mac), within the Alchemy data folder. You can create your own .tun files with the help of Manuel Op de Coul's free *Scala* software (http://www.xs4all.nl/~huygensf/scala/).

The PRESERVE button

You can 'lock in' various Master parameter settings with the PRESERVE button, so that their current values are preserved when new presets are loaded. For instance, if you are playing live with other musicians and everyone is tuned to a lower or higher reference pitch, you can adjust the fine tuning of Alchemy to match the rest of the group and then click 'PRESERVE'. This will ensure that the proper fine tuning remains in effect as you load new presets.

Parameters affected by PRESERVE are: Pitchbend Up/Down, Tuning, Coarse/Fine Tune.

Modulation

Alchemy features a unique ‘modular’ modulation system which combines ease of use with almost limitless flexibility. It’s a bit different from an ordinary synthesizer’s modulation system, but don’t worry: you can learn the basics in no time.

Working with modulators

Nearly every knob on the Alchemy interface is a **mod target**, representing a parameter that can be modulated by as many as five modulators. Whan a knob has one or more modulations assigned to it, a green mod arc is illuminated alongside the knob’s blue value arc. This confirms that the knob is a mod target, and also indicates the effective range of the modulation.

To view or edit the modulations assigned to a parameter, normally you just have to click the parameter’s knob. The parameter name is shown in the **Target** field at the top left of the Mod section; directly below it, you’ll find a **mod rack** with five slots. Click a different knob and the mod rack will be updated accordingly.

- To **select**, **create** or **change** a modulator, click a slot in the mod rack and choose from the pop-up menu that appears.
- To **undo** a modulation assignment, choose ‘None’ from the pop-up menu. The modulator will no longer affect the currently selected target (although it will still be available for assignment to other parameters).

- To **delete** a modulator entirely (so that it no longer affects any parameters), choose 'Del' from the appropriate sub-menu (e.g. to delete LFO 2, click a slot and in the pop-up menu choose 'LFO' > 'Del LFO 2').

You can also work with modulators by **right-clicking** a target knob and choosing a command from the context menu that appears.

- 'Add modulation' lets you assign a new modulator by selecting it from the appropriate sub-menu. The new modulation will appear in the first empty slot of the mod rack.
- 'Clear modulation' removes all modulations from the knob, leaving the mod rack empty.
- 'Copy modulation' places information on the clipboard about all the currently assigned modulators.
- 'Paste modulation' applies all the modulator information from the clipboard. Using the 'Copy' and 'Paste' commands, you can quickly assign the same modulations to multiple targets.

Note that the most recently clicked knob is highlighted with an illuminated spot in the center. This makes it easy to see at a glance which knob represents the current target. If you switch between sub-pages in the interface (e.g. from the Source A to the Source B sub-page), the highlight will shift to the corresponding knob on the new page, so you can quickly set up modulations to a recurring parameter such as Fine Tune on multiple sub-pages.

Types of modulator

The available modulator types are as follows:

- LFO
- AHDSR
- MSEG
- Sequencer
- Note Property
- Perform

Each one is described on its own page.

Modulation rack controls

Each slot in the mod rack has its own **ON** button (for toggling the loaded modulator on or off) and its own **Depth** control (adjustable from -100% to 100%). Double-clicking a Depth control resets it to zero.

Each slot also has a button marked **E** (for Edit). Clicking it causes the selected modulator's control panel to be displayed in the right-hand half of the MOD section. Click a different modulator's Edit button and the display updates accordingly.

Between the **E**dit button and **Depth** control, each slot also has a pop-up menu where you can assign, create or delete a ModMap.

The ModMap module is described on its own page.

Finally, a **Smooth** knob is available at the top of each mod rack. At the default value of zero, Smooth has no effect. At higher values, Smooth causes the target to respond more gradually to modulation. If you find that rapid Filter or Amp modulation causes unwanted clicks in your sound, one solution is add a small amount of smoothing via this knob. Larger values of smoothing also produce interesting effects.

Modulator knobs are different

Normally, two things happen as soon as you click a knob. First, the mod rack associated with the clicked knob is displayed at the left of the Mod section. And second, if a modulator has already been assigned to the first slot in the newly displayed rack, then its control panel appears in the right-hand half of the MOD section.

But in two cases, Alchemy prevents these things from happening. If the knob you click is located on a modulator control panel (for instance, if you've clicked LFO Rate or AHDSR Attack Time), Alchemy assumes you want to adjust the modulation of the current target instead of switching to a new target. Similarly, if you click on a modulation Depth knob in the mod rack, Alchemy assumes you want adjust the depth setting, so it keeps the current rack (and thus the knob you've just clicked) in view.

You can still view and edit modulation assignments in these cases. If a knob is located on a modulator control panel or in the mod rack, just **right-click** it to access a contextual menu with all the usual commands plus a new one: 'Edit Modulation' brings the knob's mod rack into view immediately. For more details, see the examples below.

Note that a few knobs cannot be modulated.

- The 'Smooth' knob at the top of each mod rack is not a mod target.
- You can modulate a mod depth. But you can't modulate the depth of modulation of a mod depth — which is just fine, because it would take more sentences like this one to explain it if you could!
- Knobs in the Perform section are not mod targets; they are intended as sources, rather than targets, of modulation.
- If a parameter is set via a menu selection rather than a knob (e.g. Loop mode, Filter type), then you can't modulate it.

Apart from these few exceptions, if it's a knob, you can modulate it in Alchemy. This includes additive/spectral/granular play position, AHDSR times and levels, and numerous Effects parameters. Try them and see!

Finding a modulator's targets

The mod rack shows you at a glance all the modulators that are applied to a particular target. But consider the reverse situation: how do you find all the targets of a particular modulator? This information is available via the **Target** button, which is located at the top right of the Mod section.

Clicking the Target button opens a pop-up menu listing all the modulators currently in use, and all the targets to which each modulator is assigned. For example, if you initialize Alchemy (by clicking the FILE button and choosing 'Clear' from the pop-up menu), then you can browse the Target menu to see quickly that:

- AHDSR 1 modulates Master Amp
- Velocity (a Note Property) modulates Master Amp
- No other modulations have been assigned

A very simple modulation example

1. In Alchemy's Title bar, click **FILE** and choose 'Clear' to initialise the preset. Play a note and you'll hear the familiar default sawtooth wave.

2. In the Source section, click the **Amp** knob for Source A. In the Mod section, the **Target** field will change to read 'Amp A'.

3. Click in the top slot of the rack and from the pop-up menu choose 'LFO' and then 'LFO 1'. Click the **LFO** button to display the LFO controls in the field to the right.

Notice that Source A's Amp knob is now surrounded by an illuminated green band. Read 'Working with modulators' (above) to find out why!

4. Hold a note and you'll hear a 'tremolo' effect, as Source A's amplitude is modulated by LFO 1. Change the LFO's Rate setting to adjust the speed of the effect.

5. Try adding another modulator to the rack (in the slot beneath where LFO 1 is loaded) and see what effect that has.

Up to five different modulators of any type can be assigned **to each control** in a preset.

Modulating LFO Rate

In the preceding example, we created a 'tremolo' effect by modulating Source A's **Amp** knob with LFO 1. Next, we'll modulate the Rate of this LFO with KeyFollow, so that the tremolo effect is faster for higher notes.

Note: you can adapt this example to work with other modulator parameters, such AHDSR Attack time.

1. Right-click LFO 1's Rate knob and choose 'Add Modulation' > 'Note Property' > 'KeyFollow'.

2. 'LfoRate' now appears in the Target field, and KeyFollow is listed in the first slot of the mod rack.

3. Play a variety of higher and lower notes to hear the effect of KeyFollow on the LFO Rate. If this effect is too extreme, you can reduce the Depth setting in the first slot of the mod rack.

3. You can read more about the KeyFollow modulator on the Note Property page.

Modulating modulation depth

Each modulator has its own **Depth** control, and each **Depth** control can itself be assigned a modulator.

Note: a modulator can even be set to modulate its own Depth!

1. Right-click a **Depth** control in the modulation rack.

2. From the pop-up menu that appears choose 'Edit Modulation'.

3. The **Depth** control's name will appear in the **Target** field, e.g. 'Mr1Depth' (indicating that the target is Mod rack, slot 1 Depth).

4. Add one or more modulators to the rack in the usual way.

5. Note that the Modwheel is not available directly as a modulator. Instead, the Modwheel is always linked with one of the Perform controls. If you are starting from an initialised preset, you can assign the Modwheel to control a mod Depth by choosing 'Perform' > 'Control7' as a modulator.

LFO

The **LFO** module provides a standard Low Frequency Oscillator, such as you might find in any conventional synthesizer or sampler. There are one or two unusual features, however.

- **TRIGGER**. When the **TRIGGER** button is activated, the LFO re-triggers (starts again from zero) with each new note. With **TRIGGER** deactivated, the LFO is free-running.
- **FILE**. Clicking the **FILE** button opens a pop-up menu from which LFO presets can be loaded or saved to files (*.lfo). A collection of useful presets is included (in Alchemy/Libraries/Lfo). You can also 'Copy' and 'Paste' settings between LFOs, or choose 'Clear' to initialise the module.
- **BIPOLAR**. When the **BIPOLAR** button is activated the LFO outputs both negative and positive values in each cycle (from -50% to 50%). With **BIPOLAR** deactivated, only positive values are output (from 0% to 100%).
- **Current** LFO field. Alchemy provides up to 16 LFOs (one by default, more if you create them when assigning modulators — see the Modulation page of this Manual for details). You can access each LFO's control panel by selecting its number in the Current LFO field.
- **Shape**. Click the **Shape** field to choose an oscillator waveform for the LFO. There are several categories:
    - 'Basic' contains the most familiar choices, such as 'RampUp', 'RampDown', 'Sine', 'Square', and 'Triangle'.
    - 'Basic' also contains two randomized choices. Use 'RandGlide' when you need a ***constantly fluctuating random modulator***; it reaches new random values at a speed set by the Rate control, and it glides smoothly from one value to the next. Use 'RandHold' when you need a ***stepped random modulator*** (e.g. for ***'sample and hold'*** effects); it jumps to new random values at a speed set by the Rate control, holding each value until the next jump occurs.
    - The remaining categories offer more complex shapes; it takes a bit of trial-and-error to learn which shapes are suitable for a particular purpose. The 'Serial-

Angular' category offers a variety of 'stepped' shapes, while the 'Serial-Smooth' category provides various complex rising-and-falling patterns. Finally, the 'UHF' (ultra-high frequency) category offers shapes containing multiple copies of some pattern, such that the effective modulation speed is a multiple of the speed set by the Rate control.

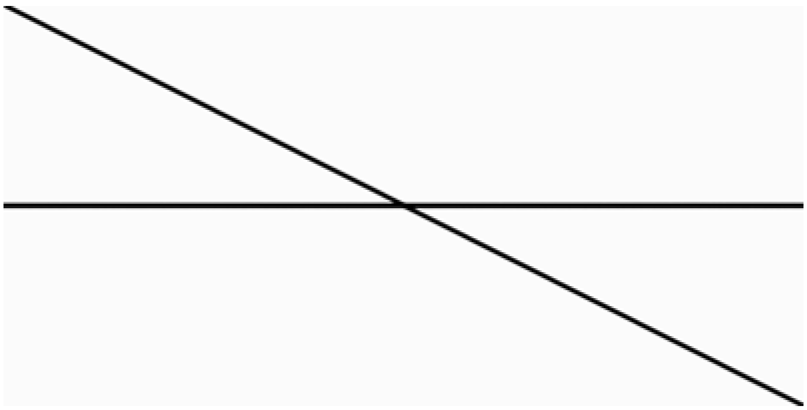
- **Rate**. Controls the LFO rate or frequency. When the **Sync** button is deactivated, the rate is adjustable over a range of 0 Hz to 220 Hz. With **Sync** activated, tempo information from the host application is used to calibrate the **Rate** control in rhythmically meaningful units (i.e. beats and fractions of a beat).
- **Sync**. Determines whether the Rate control (see above) is set in Hz (Sync off) or in Beats (Sync on).
- **Delay**. When **TRIGGER** (see above) is activated, **Delay** introduces a delay between the note-on message and the first cycle of the LFO. The delay is adjustable over a range of 0.00 seconds to 20.00 seconds. When **TRIGGER** is deactivated, **Delay** has no effect.
- **Attack**. When **TRIGGER** (see above) is activated, **Attack** applies an envelope to the LFO's output, effectively 'fading it in', so that the modulation depth steadily increases the longer a note is held. The attack time is adjustable over a range of 0.00 seconds to 20.00 seconds. When **TRIGGER** is deactivated, **Attack** has no effect.
- **Phase**. When **TRIGGER** (see above) is activated, **Phase** allows the starting point of the LFO to be adjusted from 'zero' to later in the cycle. The available range is 0.00% to 100.00%.

Note: the LFO module has no depth control. The modulation depth is adjusted using the Depth knob alongside the module's slot in the modulation rack.

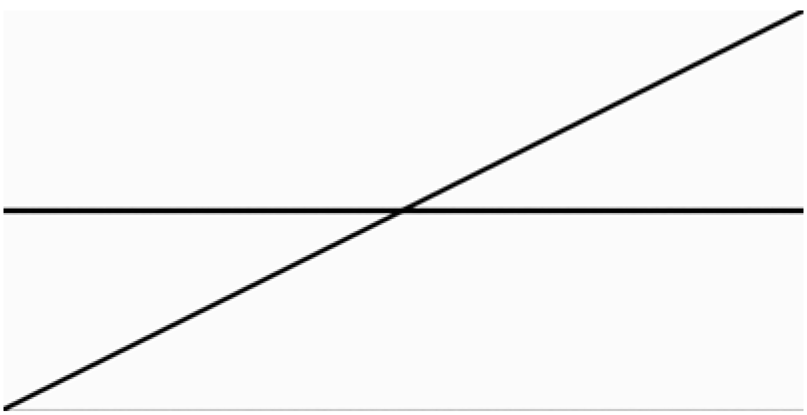
LFO shapes illustrated

Basic

RampDown

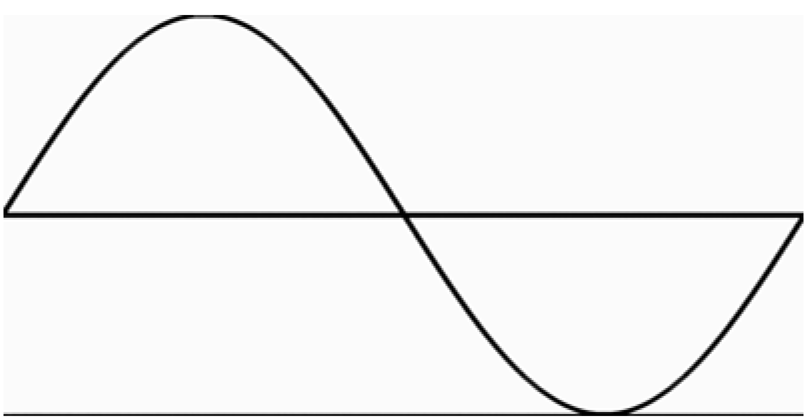


RampUp

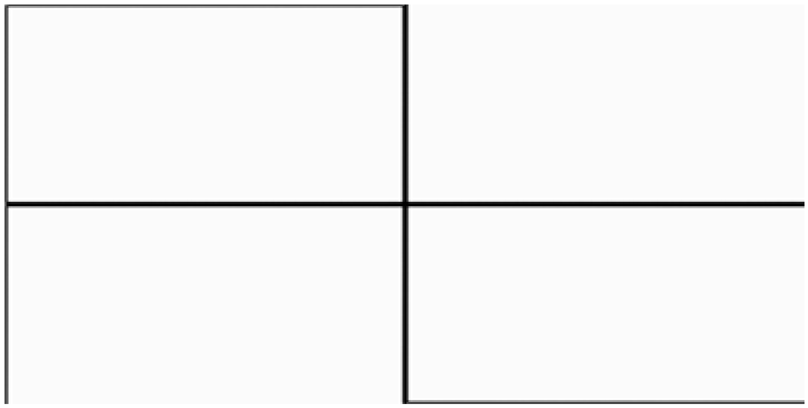


RandGlide and RandHold are described on the main LFO page.

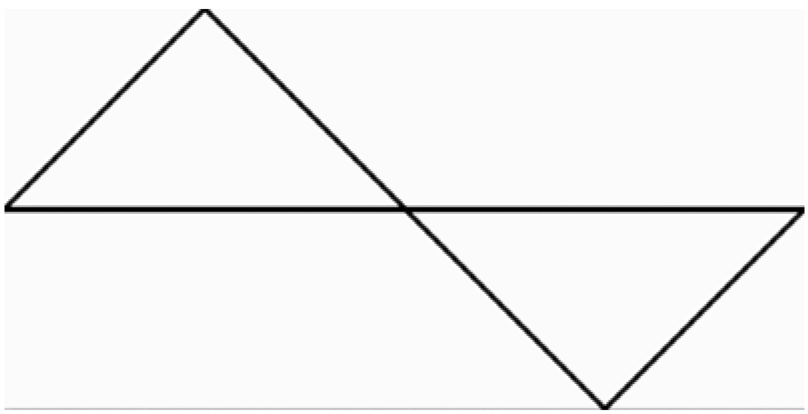
Sine



Square

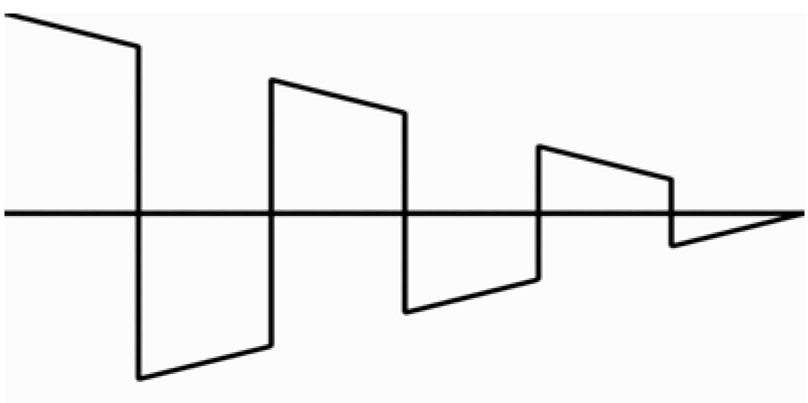


Triangle

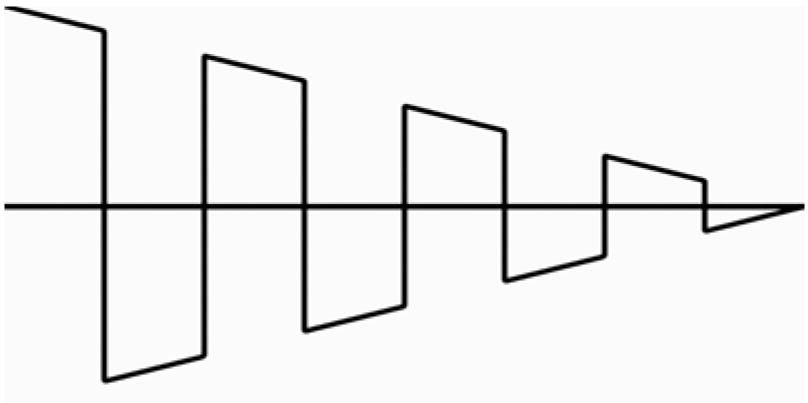


Serial-Angular

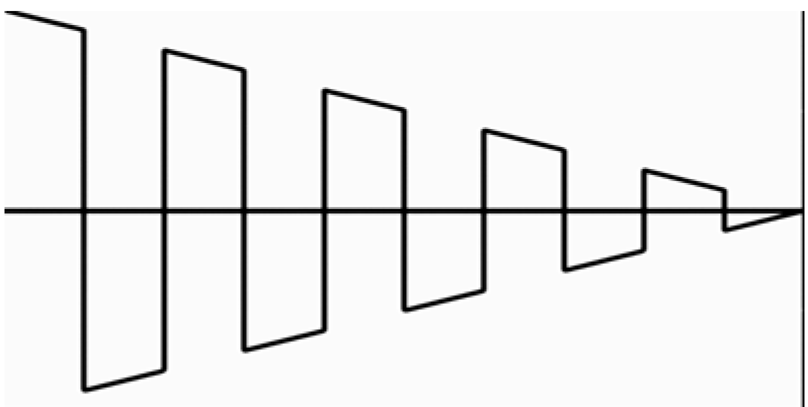
For-Sqr03



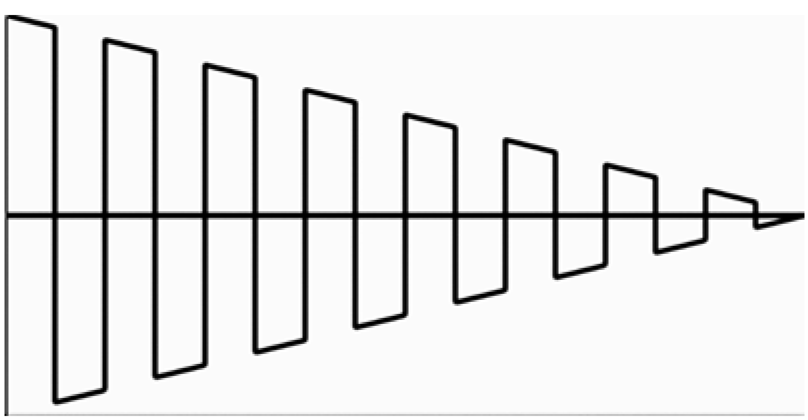
For-Sqr04



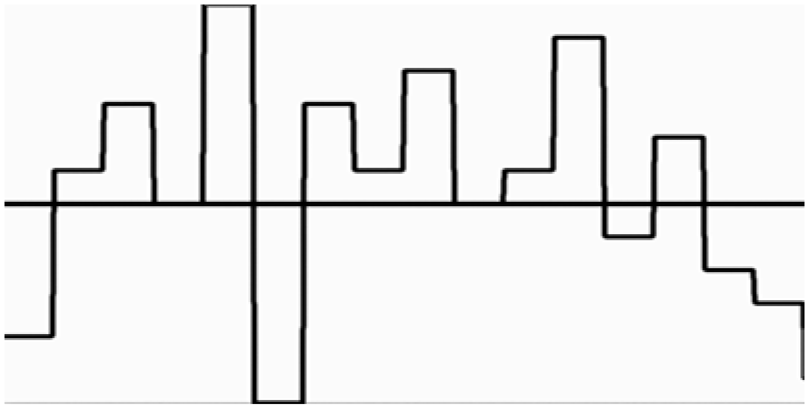
For-Sqr05



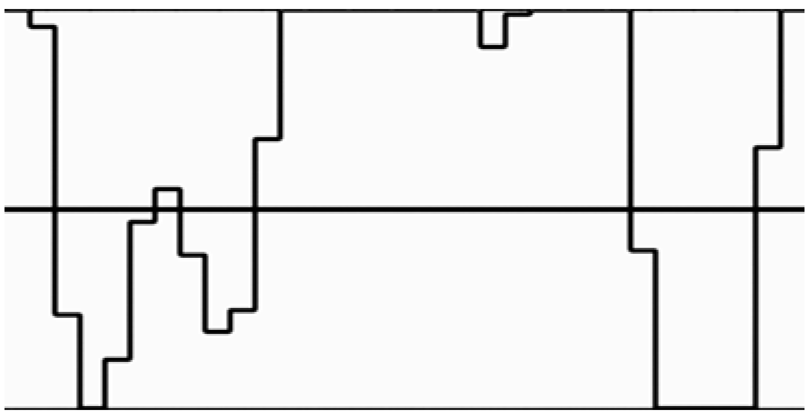
For-Sqr08



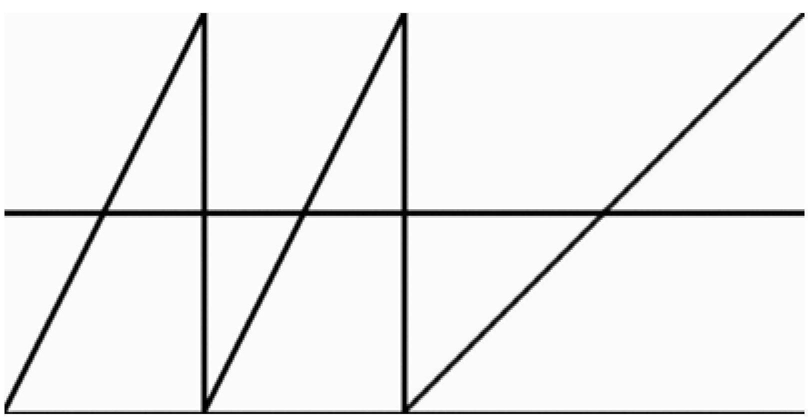
Rnd-16Step



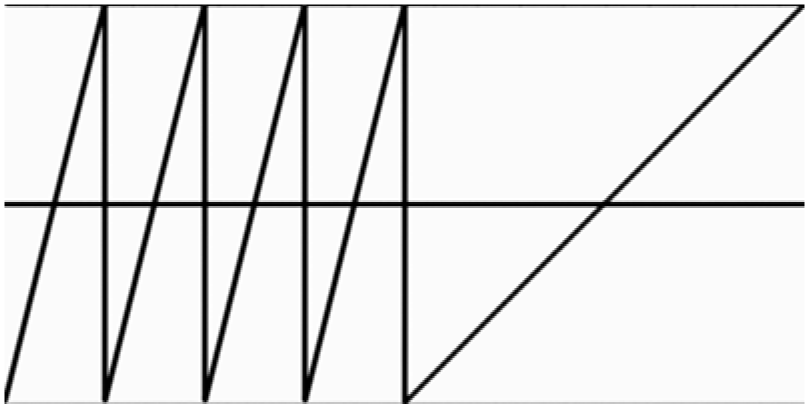
Rnd-32Step



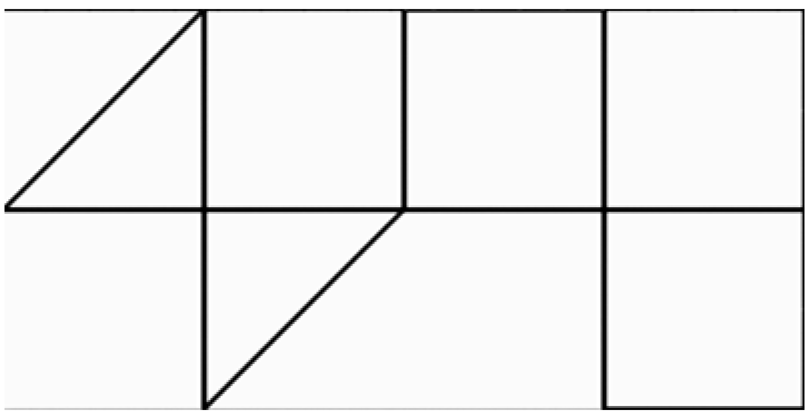
Ser-Saw2



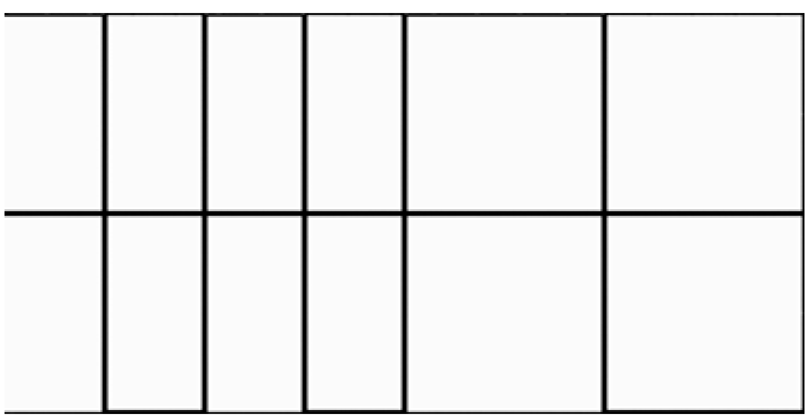
Ser-Saw4



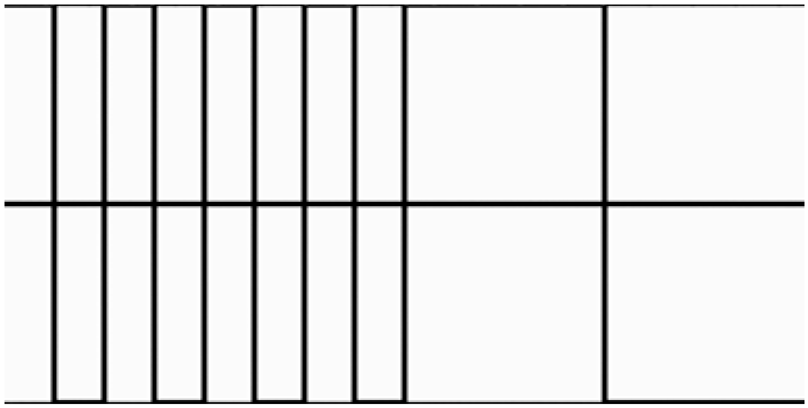
Ser-SawSq



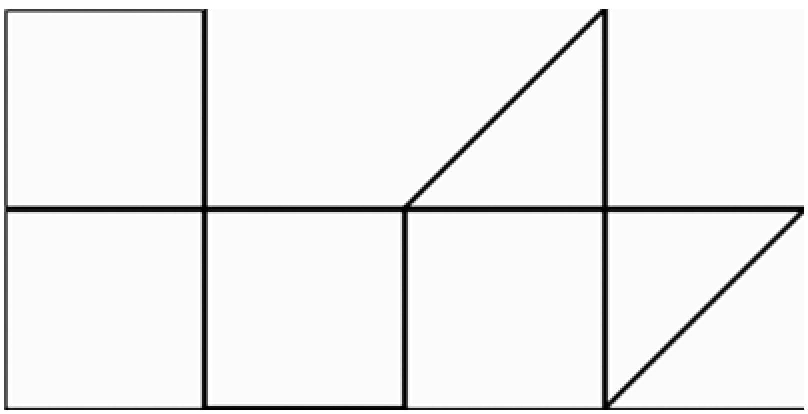
Ser-Sqr2



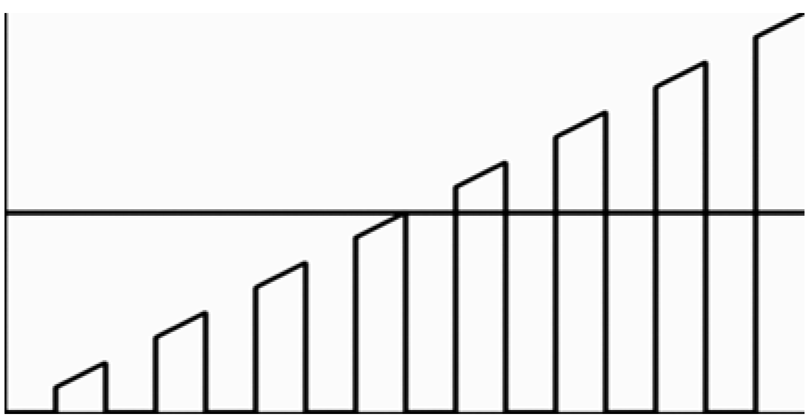
Ser-Sqr4



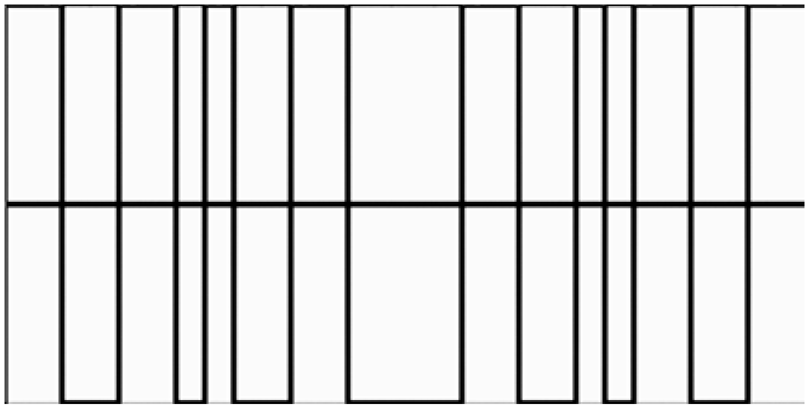
Ser-SqrSaw



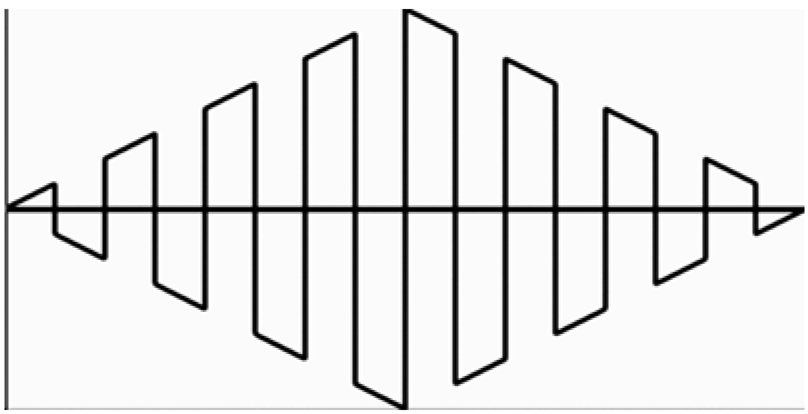
Ser-8RampUp



Ser-AM0702

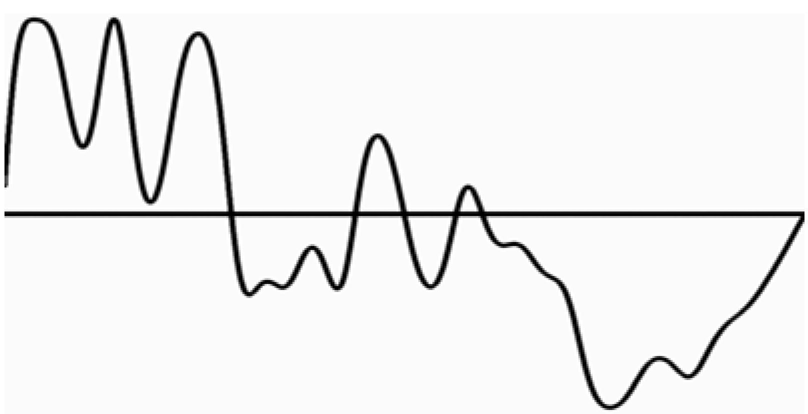


Ser-AM0Tri

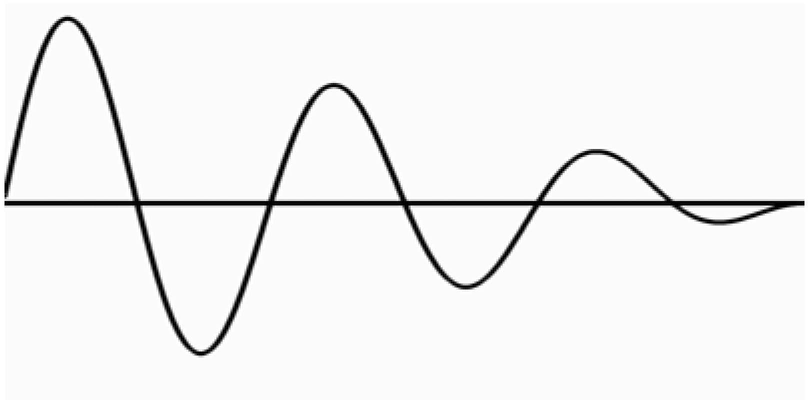


Serial-Smooth

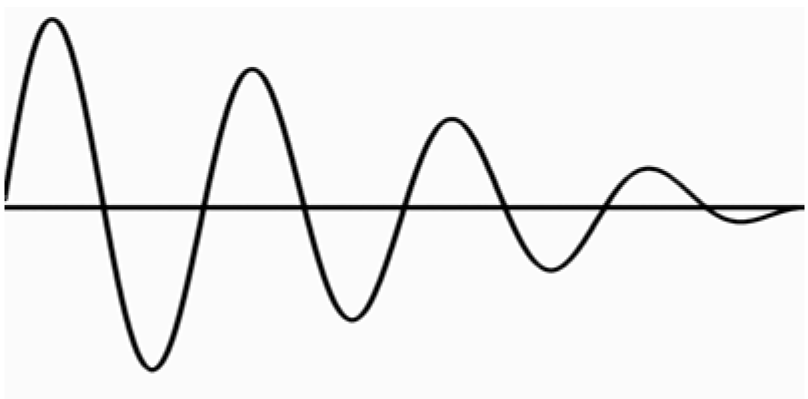
Asy-023



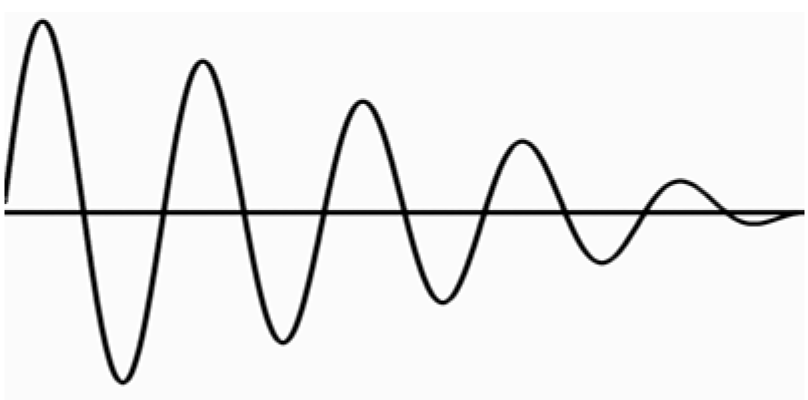
For-03



For-04

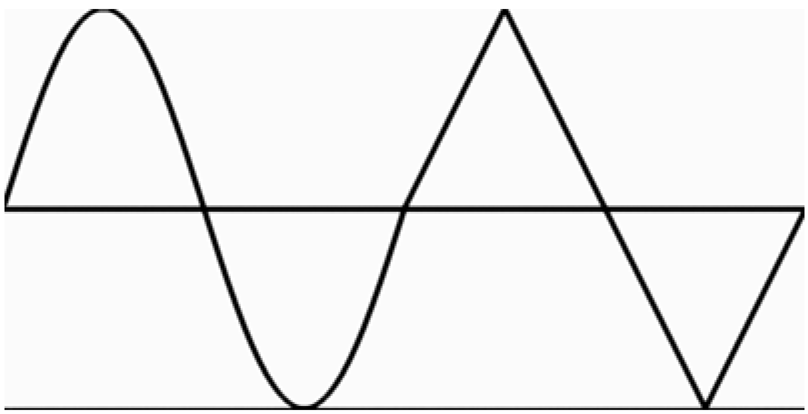


For-05

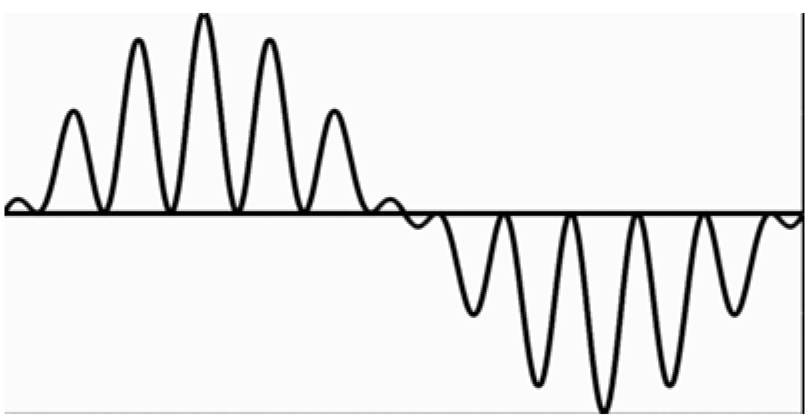


For-08

Ser-SinTri

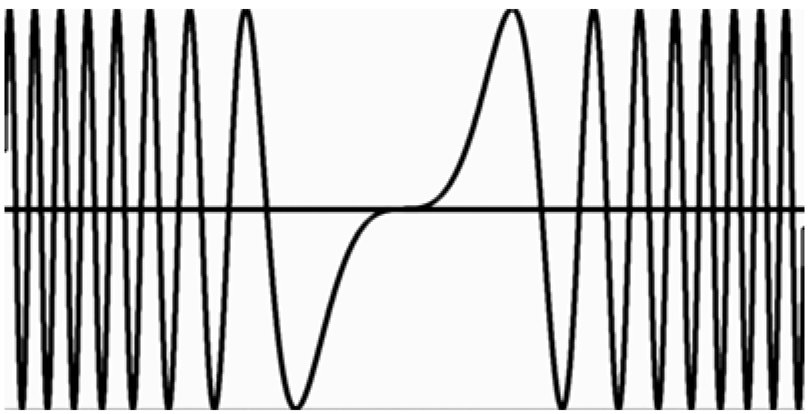


Sin-AM0112



Sin-AM0803

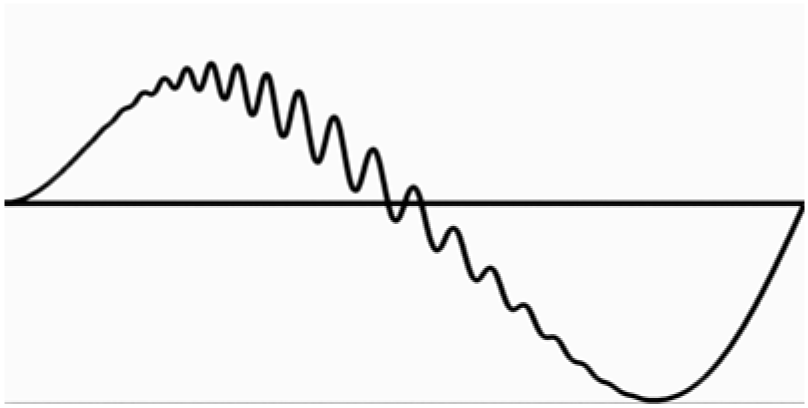
Sin-FM16



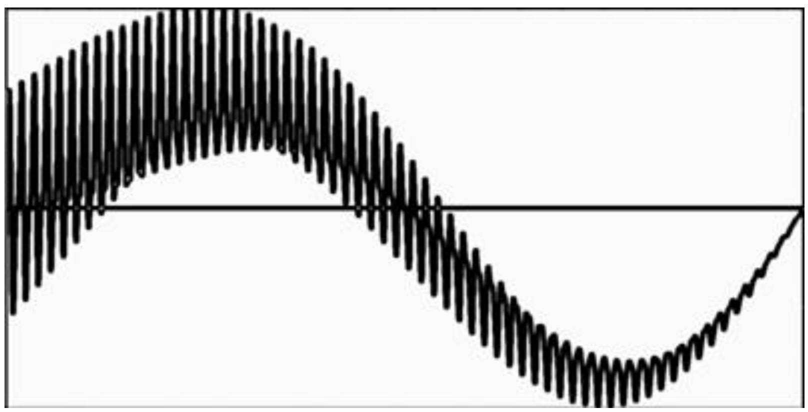
Tri-AM0501

UHF

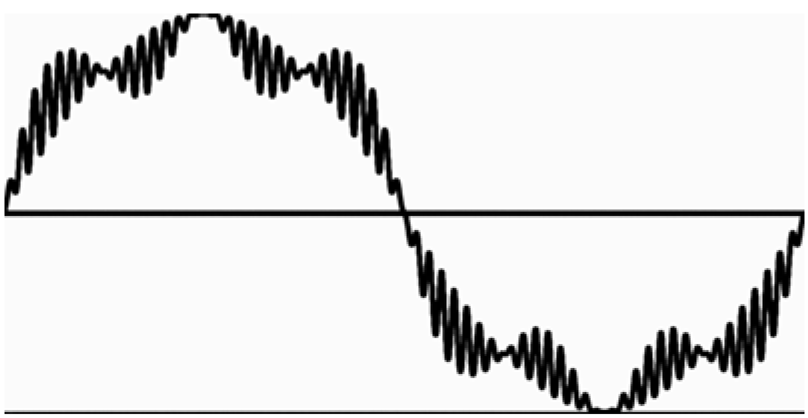
Asy-037



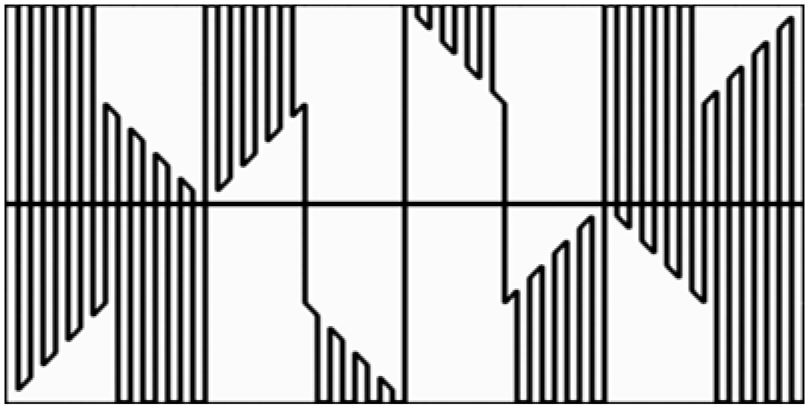
Asy-042



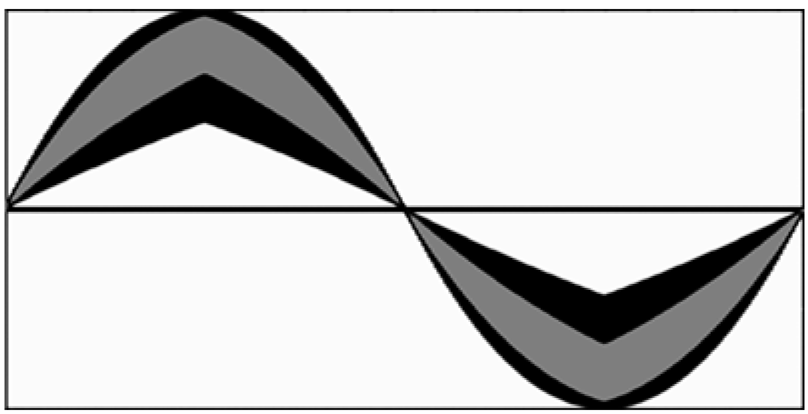
Sin-AMFM2



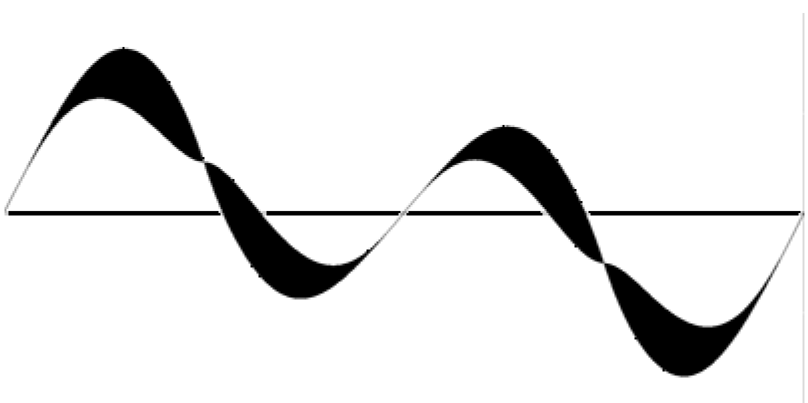
UHF-AMNS1



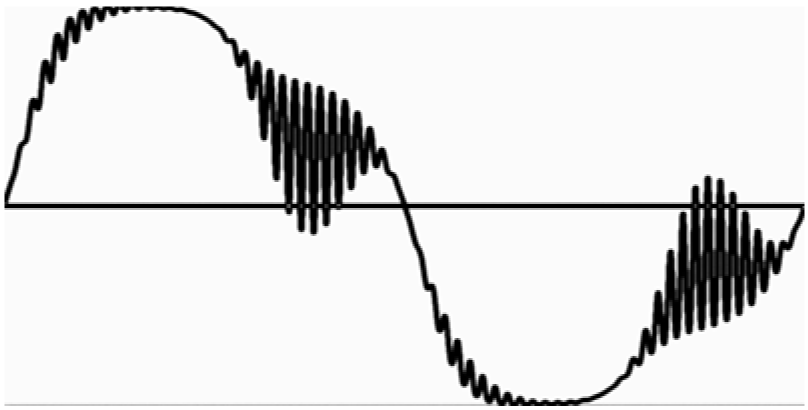
UHF-Crawlr



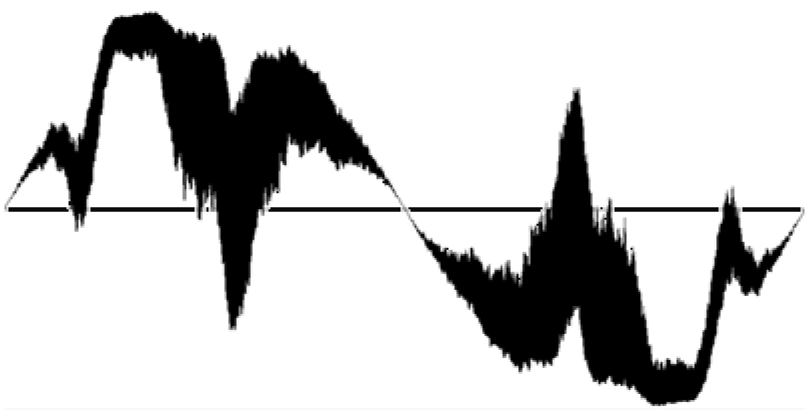
UHF-DNA



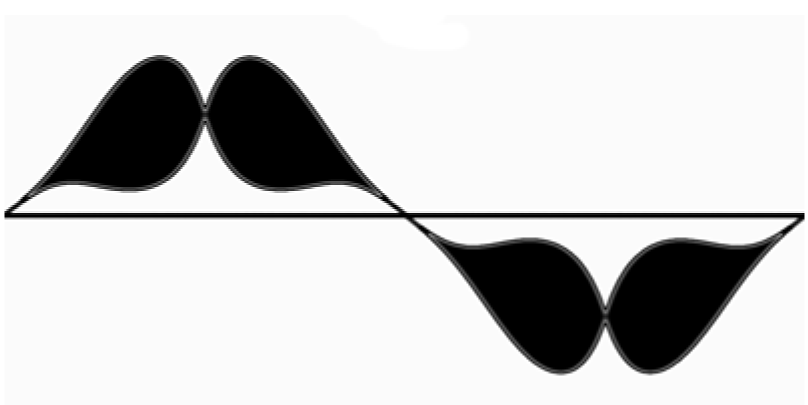
UHF-Salman



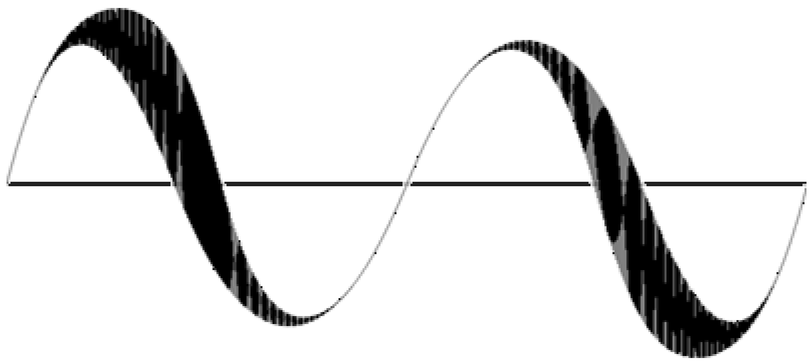
UHF-Siera



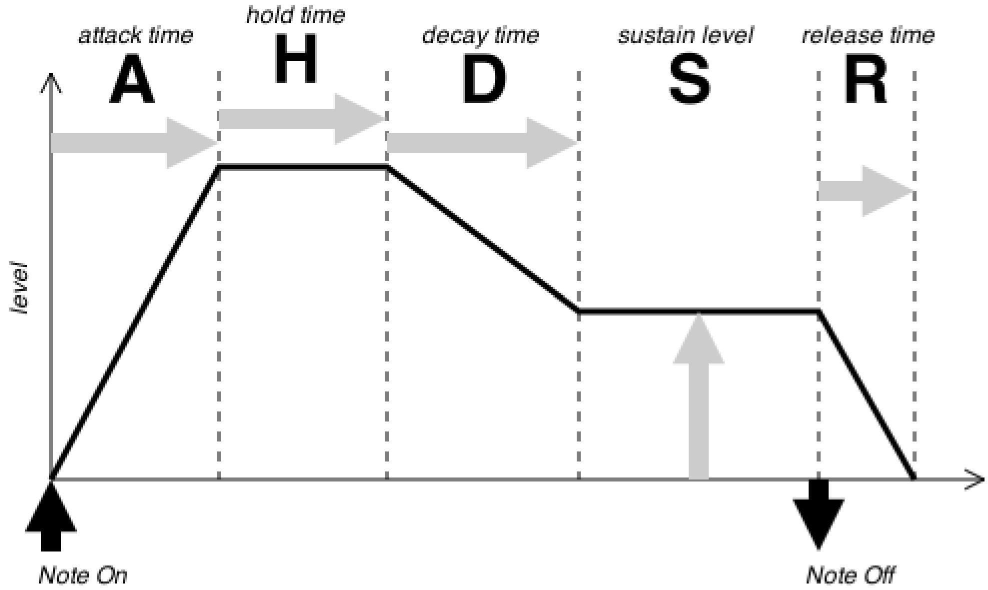
UHF-Tennis



UHF-Trump



AHDSR



The **AHDSR** module provides an envelope generator with Attack, Hold, Decay, Sustain, and Release stages. As pictured above, the A, H, D, and R stages have settable times, while the S stage has a settable level (which is maintained until a Note Off occurs). This is a fairly standard type of modulator, although the implementation in Alchemy is more versatile than most.

A description of each control follows.

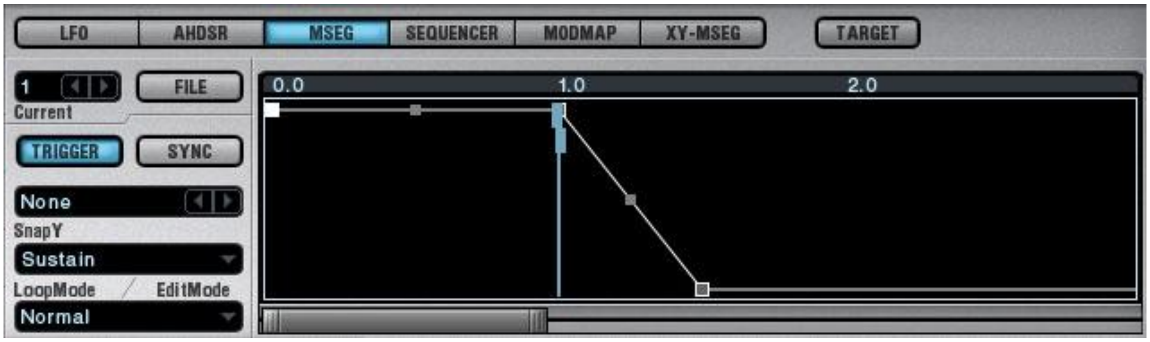
- **Current** AHDSR field. Alchemy provides up to 16 AHDSRs (one by default, more if you create them when assigning modulators — see the Modulation page of this Manual for details). You can access each AHDSR's control panel by selecting its number in the Current AHDSR field.
- **FILE**. Clicking the **FILE** button opens a pop-up menu from which envelope presets can be loaded or saved to files (*.ahd). A collection of useful presets is included (in the 'Libraries' folder within Alchemy's data directory). You can also 'Copy' and 'Paste' settings between AHDSR modules, or choose 'Clear' to initialise the module.

- **TRIGGER**. When the **TRIGGER** button is activated, the envelope re-triggers (starts again from the beginning) with each new note. With **TRIGGER** deactivated, the envelope is effectively monophonic (i.e. it triggers on the first note and persists for subsequent notes in the same legato phrase).
- **Attack**. Controls the envelope's 'attack' time, i.e. the time taken to reach peak amplitude after a note is played. From 0.00 seconds to 20.00 seconds.
- **Hold**. Controls how long peak amplitude is held before the 'decay' stage of the envelope begins. From 0.00 seconds to 20.00 seconds.
- **Decay**. Controls the envelope's 'decay' time, i.e. the time taken for the amplitude to ramp down to the 'sustain' level. From 0.00 seconds to 20.00 seconds.
- **Sustain**. Sets the envelope's sustain level, as a percentage of peak amplitude (adjustable from 0% to 100%).
- **Release**. Controls the envelope's 'release' time, i.e. the time taken for signal amplitude to ramp back down to zero. From 0.00 seconds to 20.00 seconds.

Envelope shaping

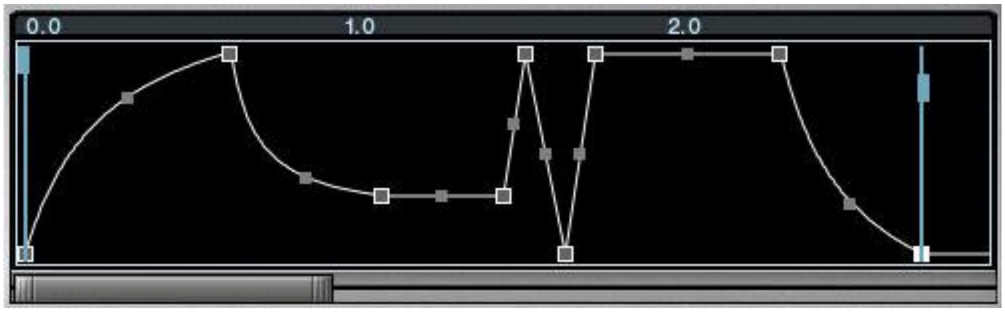
The **Attack**, **Decay** and **Release** controls each have an associated set of switches, giving you a choice of shapes for each of these stages. There are a linear shape; two **convex** shapes, with different degrees of steepness; and two **concave** shapes, again with different degrees of steepness. The different shapes produce characteristically different effects.

MSEG



The MSEG module provides a sophisticated **M**ultiple **S**egment **E**nvelope **G**enerator that allows complex modulation envelopes to be created and edited.

Any number of **breakpoints** can be added to an envelope. The envelope segments linking these breakpoints can be linear or curved. A **Sync** function allows envelopes to be anchored to a grid derived from the host application's tempo to create elaborate rhythmic patterns.



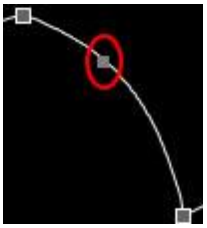
The MSEG display shows a graph of the envelope generator's output. The ruler along the top shows the time, calibrated in seconds (or in beats when Sync is activated — see below). The envelope appears as series of small squares (breakpoints) joined by lines/curves representing the different envelope segments.

Editing MSEG envelopes

Envelopes are created and edited in two basic ways: by adding, moving or removing breakpoints, and by adjusting the curve of the envelope segments between breakpoints.

- To **move** a breakpoint, simply click and drag it.
- To **add** a breakpoint, right-click (control-click) at the place where you want the new breakpoint to appear.
- To **remove** a breakpoint, right-click it (control-click).

A breakpoint's *x* value (time position) and *y* value (level) are reported together in the parameter value display when you click the breakpoint, and the displayed values update as you drag the breakpoint.
Each envelope segment has a 'handle' (a square box, slightly smaller than a breakpoint) halfway along its length, which is used to adjust the curvature of the segment.



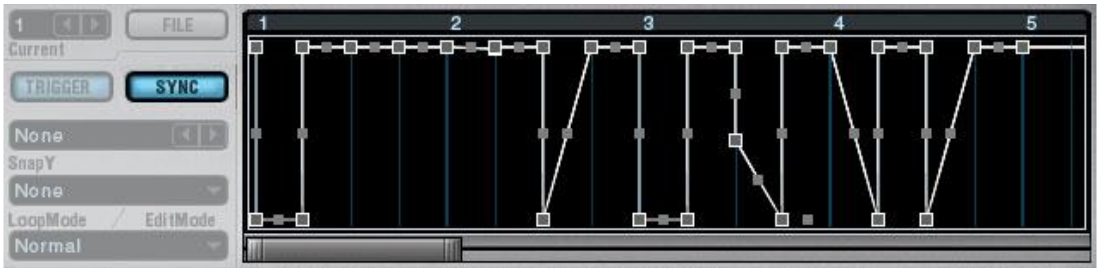
Dragging the handle upwards causes the segment to become progressively more convex; dragging it downwards causes the segment to become progressively more concave. Convex, 'flat' and concave envelope segments produce characteristically different effects.

Edit modes

The **Mode** pop-up menu allows you to choose from one of four editing modes that determine how the envelope will react to being edited.

- 'Normal'. One breakpoint at a time can be moved, by clicking and dragging. The surrounding points remain stationary.
- 'Slide'. Dragging a breakpoint also moves all subsequent points in the envelope, so that the relative distance between these points is preserved.
- 'Stretch'. Dragging a breakpoint to the left compresses earlier points and stretches later points, while dragging it to the right stretches earlier points and compresses later points. In either case the total length of the envelope is preserved.

Sync

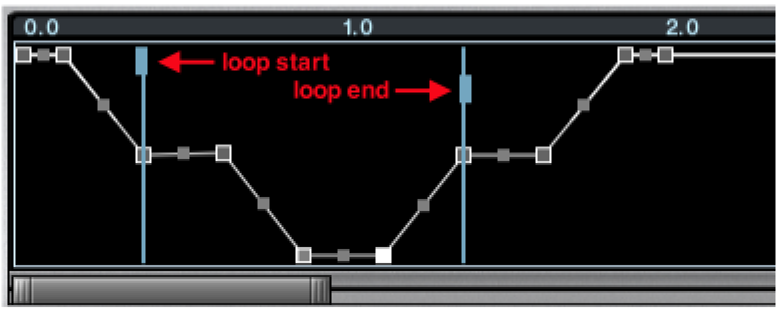


When the **Sync** button is activated, the time ruler along the top of the MSEG display is recalibrated in beats (calculated from the host application's current tempo) and a grid of vertical lines (spaced at quarter-beat intervals, or less frequently when zoomed out) is superimposed. Breakpoints 'snap' to these grid lines when dragged or created, making it easy to create precisely-aligned rhythmic envelopes.

Loop markers

Two pale blue vertical lines also appear in the MSEG display, each with a small rectangular 'handle'. These are the envelope loop markers.

> ***Note: if 'None' is selected in the Loop pop-up menu (see below) the envelope has no loop, and the loop markers are consequently not displayed.***



The marker with the higher, leftward handle controls the loop **start** point. The marker with the lower, rightward handle controls the loop **start** point.

The loop markers are moved by simply dragging them in the desired direction. For obvious reasons, the loop start marker cannot be moved to the left of the loop end marker, and the loop end marker cannot be moved to the right of the loop start marker. Loop markers always snap to the nearest breakpoint.

The loop modes, accessible from the **Loop** pop-up menu, are as follows:

- 'None'. Looping is disabled.
- 'Continuous'. The looped section plays continuously in a forward direction while a note is held, and goes on looping after the note is released.
- 'Sustain'. The loop section is played while a note is held. When the note is released the remainder of the envelope plays.
- 'Forward/Back'. Like 'Continuous' except that the looped section is played alternately forwards and backwards.

Other controls and parameters

- **SnapY** quantizes the breakpoint levels (or *y* values), limiting them to exact fractions of the available range. For instance, a SnapY setting of ‘1/3’ means that breakpoint levels will snap to the values 0, 1/3, 2/3, and 1 when dragged. A SnapY setting of ‘None’ turns quantization off and allows you to set breakpoint levels freely.

  Note that the SnapY setting does not move existing breakpoint levels into alignment with quantized positions; it only affects how breakpoints respond when created or dragged.

- **Current** MSEG field. Alchemy provides up to 16 MSEGs (two by default), and you can access each MSEG’s control panel by selecting its number in the Current MSEG field.

- **FILE**. Clicking the **FILE** button opens a pop-up menu from which you can **Load**/**Save** MSEG presets from/to files (*.mse). A collection of useful presets is included (in the ‘Libraries’ folder within Alchemy’s data directory). You can also **Copy** and **Paste** settings between MSEGs, or choose **Clear** to initialise the module.

- **TRIGGER**. When the **TRIGGER** button is activated, the envelope re-triggers (starts again from the beginning) with each new note. With **TRIGGER** deactivated, the envelope is effectively monophonic (i.e. it triggers on the first note of each legato group and persists for subsequent notes).

Example: Modulating pitch

1. First, initialise the preset (by clicking **FILE** in Alchemy’s title bar, and choosing ‘Clear’).

2. In the Master section, click the **Coarse-Tune** knob, and assign ‘MSEG 1’ as its modulator.

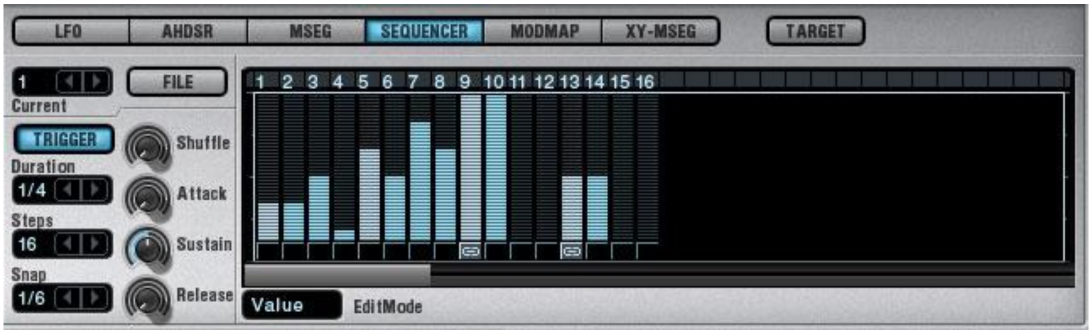
3. In the modulation rack, set MSEG 1’s **Depth** control to 24 semis.

4. If the MSEG module’s control panel is not already in view, click the **MSEG** button at the top of the Modulation section to access it. From the SnapY pop-up menu, choose ‘1/24’.

Now an MSEG envelope can be used to control Alchemy’s pitch in semitone increments, over a two-octave (i.e. 24-semitone) range.

Sequencer

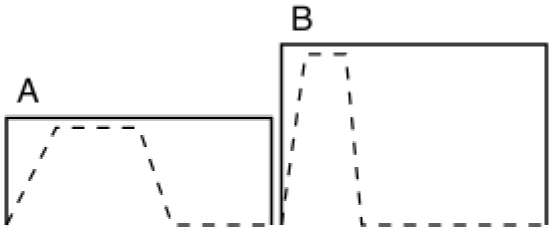
The Sequencer module provides a powerful step-based modulator that synchronizes with the host tempo and can be programmed with patterns of up to 128 steps. In addition to the level of each step, its groove/shuffle and its envelope can be controlled, both globally for the whole pattern and locally per step.



The step editor at the right-hand side of the Sequencer module is fairly self-explanatory. Steps are numbered from left to right across the top of the display. In the usual **Values mode**, the value of each step is represented by a vertical bar (or by a totally empty column when the value is zero).

- Drag a bar up or down to adjust its value, or click directly at the desired height.
- Set the values of multiple bars by simply dragging left or right across them.
- Right-click (or control-click) a bar to set it to zero.
- You can 'tie' one step to the next by shift-clicking below it. A small 'chain-link' symbol appears below each tied step. When two or more steps are tied together, they behave like one longer step.

The step editor can be switched to two additional modes besides the usual Values mode. In **Length mode**, you can edit the step lengths to create a pattern of longer and shorter envelope shapes (so that the attack, sustain, and release stages fill a larger or smaller portion of the fixed step duration). These lengths are combined with the Sequencer's overall **Attack**/**Sustain**/**Release** settings to determine the envelope shape of each step. In the following illustration, Step B has a greater *value* than Step A, but Step A has a greater *length* than Step B.



When the step editor is in **Swing mode**, finally, you can create variations in the timing of the steps. Each swing value ranges from 0 to 2; the middle value of 1 represents normal timing, while smaller values play earlier and larger values play later than normal. These swing values are combined with the overall timing pattern determined by the Sequencer **Shuffle** control.

The remaining Sequencer controls are as follows:

- **FILE** button — click to access a pop-up menu with the following commands:
    - **Save** stores the current Sequencer configuration (step values, lengths, and swing settings; plus the settings of the remaining Sequencer controls) in a new disk file (*.seq), while **Load** applies the configuration stored in an existing disk file.
    - **Copy** stores the current Sequencer configuration on the clipboard, and **Paste** applies the configuration currently found on the clipboard.
    - **Clear** resets the Sequencer controls and pattern to their default settings.
    - **Randomize** applies random offsets to the Shuffle, Attack, Sustain, and Release settings (described below).
    - The **Import** functions are described in a separate section below.
- **Current** Sequencer field — Alchemy provides up to 16 Sequencers, and if you are working with a preset that uses more than one, you can access each Sequencer's control panel by selecting its number in the Current Sequencer field.
- **TRIGGER** button — When TRIGGER is activated (lit) the sequencer pattern plays from the beginning with each MIDI note-on. With TRIGGER deactivated (unlit), the Sequencer runs continuously.

    Note that the Sequencer module always play in sync with the host tempo, regardless of the TRIGGER setting.

- **Duration** determines the duration of each and every step in the pattern, expressed as a fraction of a beat. (Assuming a quarter-note beat, '1/2' produces eighth note steps and '1/4' produces sixteenth note steps.)
- **Steps** determines the length of the Sequencer pattern (number of steps, from 1 to 128)
- **Snap** quantizes the step values, limiting them to exact fractions of the available range. For instance, a Snap setting of '1/2' means that bars will snap to the values 0%, 50%, and 100% when dragged. A Snap setting of 'None' turns quantization off and allows you to set step values freely.

Note that the Snap setting does not move existing step values into alignment with quantized positions; it only affects how step values respond when you move them by clicking and dragging in the step editor.

- **Shuffle** lets you create various ‘swing’ effects. Setting a Shuffle value greater than 0% increases the duration of the odd-numbered steps (1, 3, 5, …) and decreases the length of the even-numbered steps correspondingly.
- **Attack** determines the amount of time each step takes to reach its peak level.
- **Sustain** determines the amount of time each step is held at its peak level.
- **Release** determines the amount of time each step takes to fall from its peak level.

Importing from a MIDI file

Alchemy is capable of extracting information from a short MIDI file and applying it to patterns in the step editor. Specifically:

- It can extract *velocity* data and set the step values to match. If the MIDI file consists of notes of equal duration (e.g. a succession of eighth notes, or a succession of sixteenth notes), then every step in the resulting pattern will have an associated non-zero value. If the MIDI file consists mainly of notes of equal duration with occasional gaps (e.g. a succession of eighth notes with occasional eighth rests), then the gaps will be represented by step values of zero. If the MIDI file has more irregular timing, or if it consists of chords rather than single notes, then the results of this process will be less predictable and usually less useful.
- It can extract *groove data* (timing inflections) and set the step swing values to match. If the MIDI file consists of nearly equal durations (e.g. eighth notes or sixteenth notes with timing inflections), this process yields meaningful results.
- It can extract *note data* (pitches) and set the step values to match. The MIDI file should consist of equal durations with no gaps, and it should be limited to single pitches between a low C and a C two octaves higher (e.g. C1 through C3); the low C corresponds to a step value of zero. (Because the pitch range is always two octaves, which is equivalent to 24 semitones, you should use the sequencer to modulate pitch with a depth of 24 semitones in order to reproduce the pattern of notes in the original MIDI file.)

Three import commands in the Sequencer FILE pop-up menu make use of these capabilities.

- **Import Velocity** — sets step values based on extracted velocity data and swing values based on extracted groove data.
- **Import Note** — sets step values based on extracted note data and swing values based on extracted groove data.
- **Import Groove** — sets swing values based on extracted groove data.

Note Property

Several properties of incoming MIDI note data, as well as values generated per-note by Alchemy, are available as modulation sources.

> ***Note: unlike the other modulators, Note Property has no control panel to display.***

Click a slot in the modulation rack and choose 'Note Property' from the pop-up menu. A sub-menu appears, from which the following options are available:

- 'Velocity'. Modulation based on the velocity values of incoming MIDI note data.
- 'KeyFollow'. Modulation based on incoming MIDI note numbers (i.e. the modulation value increases as you play higher pitches on your MIDI keyboard controller). This is a bipolar source, with C3 corresponding to zero.
- 'ChanAftertouch'. Modulation based on channel aftertouch data.
- 'PolyAftertouch'. Modulation based on poly after touch data.

> ***Note: Channel aftertouch is transmitted by many MIDI controllers; it consists of one variable stream of values per MIDI channel. Poly aftertouch, which consists of a variable stream of values per individual note, is a much less common feature. If your controller doesn't produce poly aftertouch (or channel aftertouch, for that matter), you can probably still create data of this type directly in your sequencer software.***

- 'Speed'. Modulation based on the elapsed time between notes (e.g. a progressively slower sequence of notes results in progressively greater modulation values).
- 'Held'. A modulation signal that rises to full-scale immediately at note-on and falls to zero immediately at note-off.
- 'FlipFlop'. A modulation signal that is alternately full-scale and zero on successive notes.
- 'FlipFlop2'. LikeFlipFlop, but the value reverses every two notes: zero, zero, full, full, repeat.

> ***Note: The FlipFlop modulators can be used together to create a 'round-robin' involving all four of Alchemy's Source modules. With the Morph mode set to 'Morph XY' or 'XFade XY', set the Morph X and Y knobs both to 0%; then modulate X with FlipFlop and Y with FlipFlop2 (or vice***

versa). (See the [Morph](#) page for information on morphing and cross-fading.)

- 'Random1' – 'Random4'. Modulation based on a fixed random value per note. This is a unipolar source with values ranging from zero through full-scale. The four sources of this type are randomized independently of one another.
- 'PitchBend'. Modulation based on MIDI pitchbend messages. This is a bipolar source with values ranging from negative full-scale through positive full-scale.
- 'Max'. Modulation based on a constant full-scale value.

Perform

When you click a slot in the modulation rack and choose the ‘Perform’ sub-menu, you can select from a list of sixteen different modulation sources.

These correspond to the sixteen different controls (eight assignable knobs, two X/Y control squares, four envelope knobs) available in Alchemy’s Perform section. For more information on working with these controls, see the Performance controls page.

The available modulations sources are:

- ‘Control1’ – ‘Control 8’. Assignable knobs, numbered 1–8.
- ‘XYPad1X’. The X (horizontal) axis of the left-hand XY control square.
- ‘XYPad1Y’. The Y (vertical) axis of the left-hand XY control square.
- ‘XYPad2X’. The X (horizontal) axis of the right-hand XY control square.
- ‘XYPad2Y’. The Y (vertical) axis of the right-hand XY control square.
- ‘MAttack’, ‘MDecay’, ‘MSustain’ and ‘MRelease’. The **Attack**, **Decay**, **Sustain** and **Release** knobs.

ModMap

A **ModMap** is not a modulator. Instead, its purpose is to **process the output of a modulator**, mapping the original values to new ones before they are applied to a particular modulation target. With ModMaps, you can create 'curved' velocity responses, scale the volume of each Source across the keyboard, 'quantize' the pitch response to random-LFO modulation so it adheres to the steps of a scale, and much more.

This page is organized in three parts. First, it explains how to assign a ModMap to a modulation in one of Alchemy's mod racks. (Please see the Modulation page for a broader discussion of the mod rack system.) Second, it provides a number of 'recipes' for ModMaps that perform common functions; you can use these as-is or adapt them for your needs. And third, it offers a more complete explanation of how ModMaps work — the details you'll want to know when you design ModMaps of your own.

Assigning a ModMap

The **modmap field** at the far right of each mod-rack slot displays the number of the ModMap, if any, that currently applies to the modulation set up in that slot. A dash '–' indicates that no ModMap has been assigned.

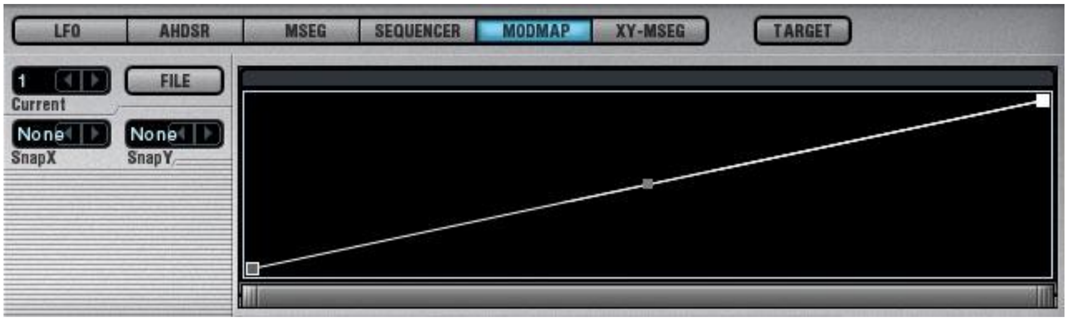
- To **select**, **create**, or **change** a ModMap, click in the modmap field and choose from the pop-up menu that appears.
- To **undo** a ModMap assignment, chose the dash '–' from the pop-up menu. The ModMap will no longer affect the modulation set up in this mod-rack slot (although it will still be available for assignment to other modulations).
- To **delete** a ModMap entirely (so that it no longer applies to any modulations), choose 'Del ModMap' from the pop-up menu.

ModMap recipes

These examples illustrate typical uses of the ModMap feature. For an explanation of why each ModMap works the way it does, read *the details* at the bottom of this page.

The default ModMap

Newly initialized presets in Alchemy have a single ModMap shaped as follows.



Each additional ModMap that you create begins with this same default shape — a straight line increasing from 0.00 at the far left to 1.00 at the far right. This **default ModMap does nothing** — it 'maps' modulations values by passing them through unchanged.

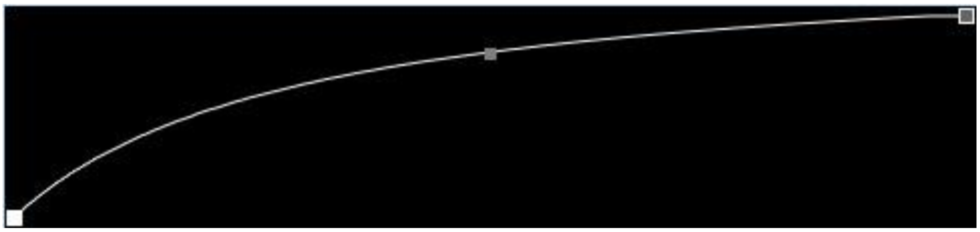
Although it doesn't do anything useful on its own, the default ModMap is a starting point for all the recipes that follow.

Velocity curve

- Initialize Alchemy by choosing the 'Clear' command in the Title bar's FILE menu. By default, an initialized preset has Velocity assigned as a modulator to Master Amp, so when you strike keys harder or softer (on a velocity-sensitive MIDI keyboard), notes play louder or softer.

- Click on the Master Amp knob to bring its mod rack into view in the MOD section. You'll see two modulators loaded in the modulation rack: 'AHDSR 1' in the first slot, 'Velocity' in the second. Click the dash '–' in the modmap field at the far right of the 'Velocity' slot, and choose '1' from the pop-up menu. This applies the default ModMap to Velocity modulation of Master Amp, and it brings the modmap controls and editor into view in the right-hand half of the Mod section. Play a few notes on your MIDI keyboard to confirm that the velocity response is unchanged by the default ModMap.

- The default ModMap (pictured above) consists of a single segment with a breakpoint at each end and a handle in the middle. Click on this handle and drag upwards. The segment will curve and become convex. Play a few more notes on your MIDI keyboard and notice the effect of the convex velocity curve: notes you strike with medium force play louder than they did with the default curve.



- Now click on the same mid-segment handle and drag downwards until the segment curves in the opposite direction, becoming concave. Play a few more notes on your MIDI keyboard and notice the effect of the concave velocity curve: notes you strike with medium force play softer than they did with the default curve.

Scaling volume across the keyboard

- Initialize Alchemy by choosing the 'Clear' command in the Title bar's FILE menu. From the Source All sub-page, Load one sample (or multi-sample) into Source A, and load a contrasting sample (or multi-sample) into Source B. By default, there is a 50% crossfade between these sources, so you should hear a balanced mix of the two samples across the entire keyboard. (For details about loading samples, see the Source page.)

> ***Note that Alchemy plays loaded samples in Granular mode by default. For ordinary sample playback, you may prefer to switch Sources A and B from Granular to Sampler mode. See the Granular page for details.***

- Suppose that we want both Sources to play at full strength from the bottom of the keyboard up through approximately C4, and that above this point we want Source A to remain at full strength while Source B gets softer as we go further up the keyboard. Since we want volume scaling on Source B only, we will operate on the Source B Amp knob rather than the Master Amp knob. (The Source B Amp knob is accessible from either the Source All sub-page or the Source B sub-page.)

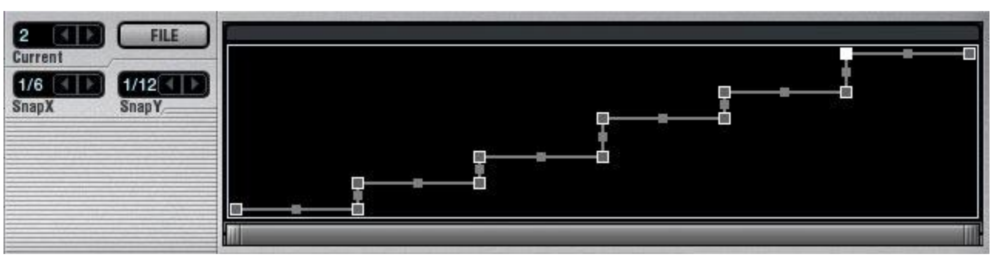
- Click the Source B Amp knob to bring its mod rack into view in the Mod section. In the first slot of the mod rack, choose 'Note Property' > 'KeyFollow'. Now the amplitude of Source B is modulated according MIDI note number, but this modulation does not yet have the shape we want across the keyboard. Currently, the lowest notes are the softest, notes in the middle of the keyboard are medium-loud, and notes at the top are the loudest. Now we will use a ModMap to reshape the response of Source B Amp to modulation by KeyFollow.

- At the far right of the first slot of the Source B Amp mod rack, click the dash '–' in the modmap field and choose '1' from the pop-up menu. This applies the default ModMap to KeyFollow modulation of Source B Amp, and it brings the modmap controls and editor into view in the right-hand half of the Mod section. Play a few notes on your MIDI keyboard to confirm that the KeyFollow response is unchanged by the default ModMap.

- In order to produce the desired KeyFollow response, edit the ModMap shape. The required shape is pictured below. Raise the leftmost breakpoint up to a level of 1.00. Create a new breakpoint by right-clicking (control-clicking) somewhat more than halfway from left to right; leave the level of the new breakpoint at 1.00. Finally, drag the rightmost breakpoint down to a level of 0.00. Now Source B Amp should respond appropriately as you play across the keyboard.

Quantizing pitch modulation to a scale

- Initialize Alchemy by choosing the 'Clear' command in the Title bar's FILE menu. Click the Master Coarse Tune knob to bring its mod rack into view in the Mod section.
- In the first slot of the mod rack, chose 'LFO' > 'LFO 1', and reduce the modulation Depth to '12.0 semis'. Then adjust the LFO 1 settings as follows: Shape = 'RandHold', Rate = '1/2 beats', BIPOLAR off. Play and hold a note to confirm that the pitch changes twice per beat.
- At this point, the pitch values range freely within the specified 12.0 semitone range. The ModMap pictured below will map the LFO values to steps of a major pentatonic scale.



- Use the ModMap editor's SnapX and SnapY functions to create the desired ModMap shape more easily. Set SnapX = 1/6 and SnapY = 1/12. In between the original first and final breakpoints, add ten new ones at the following **X, Y** positions:
    - 1/6, 0/12
    - 1/6, 2/12
    - 2/6, 2/12
    - 2/6, 4/12
    - 3/6, 4/12

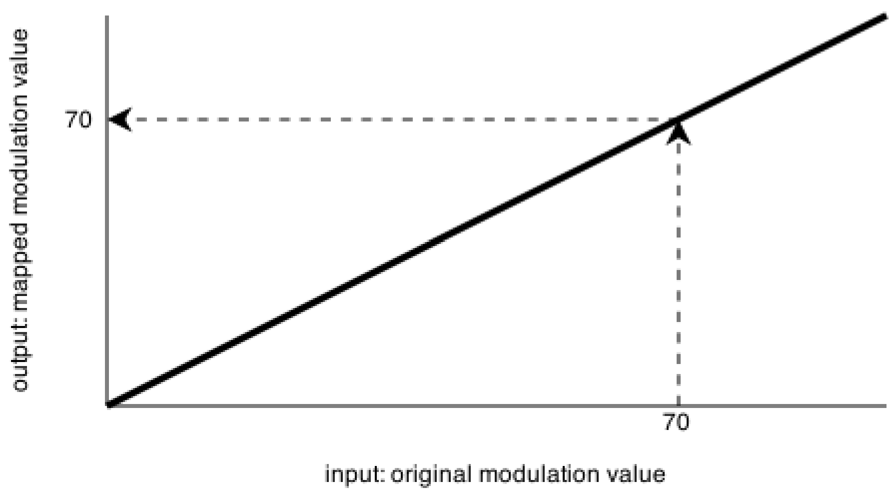
- 3/6, 7/12
- 4/6, 7/12
- 4/6, 9/12
- 5/6, 9/12
- 5/6, 12/12

Note that the Y values here reflect the interval pattern of the major pentatonic scale.

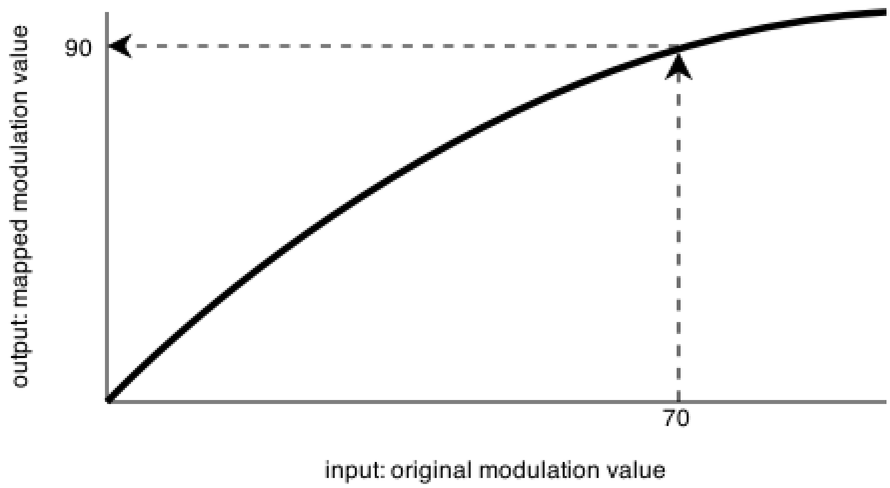
The details: how ModMaps work

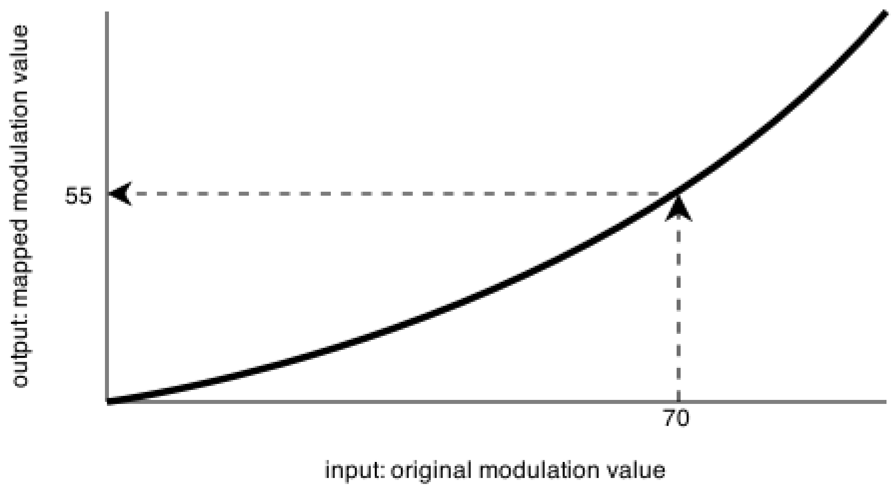
A ModMap processes the output of a modulator, mapping the original values to new ones before they are applied to a particular modulation target. The mapping is defined by the ModMap's graphical shape, which represents a **transfer function**. The x (horizontal) axis represents the range of original modulation values, from 0.00 to 1.00, while the y (vertical) axis represents the range of mapped modulation values, also from 0.00 to 1.00. To see how a modulation value will be affected by the ModMap, look up the original value along the x axis; the corresponding y value determines the output of the mapping.

- ***The default ModMap*** does nothing, because the output is identical to the input.

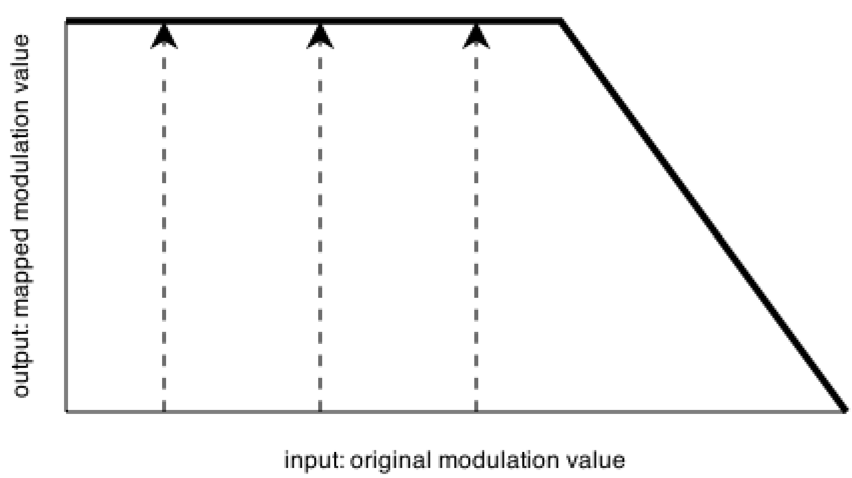


- A ***convex ModMap*** maps the middle range of inputs to values that are higher than the default output, while a ***concave ModMap*** maps the same range to values that are lower than the default output.

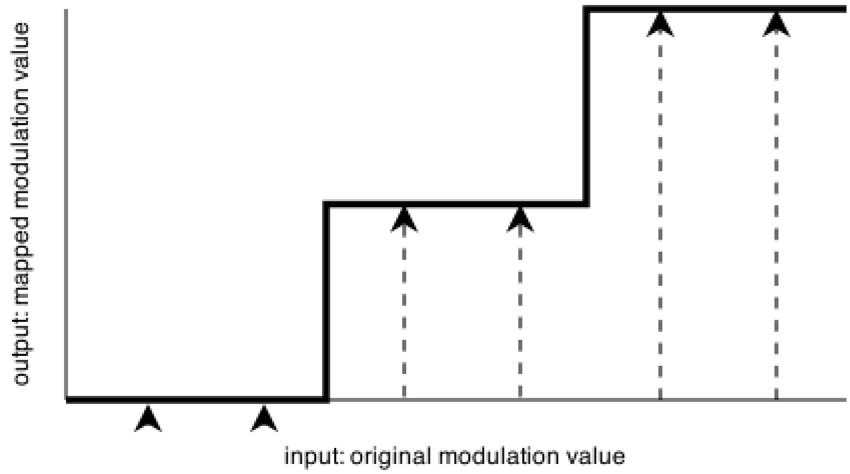




- A ***ModMap with a flat (horizontal) region*** maps a range of inputs to a single output.



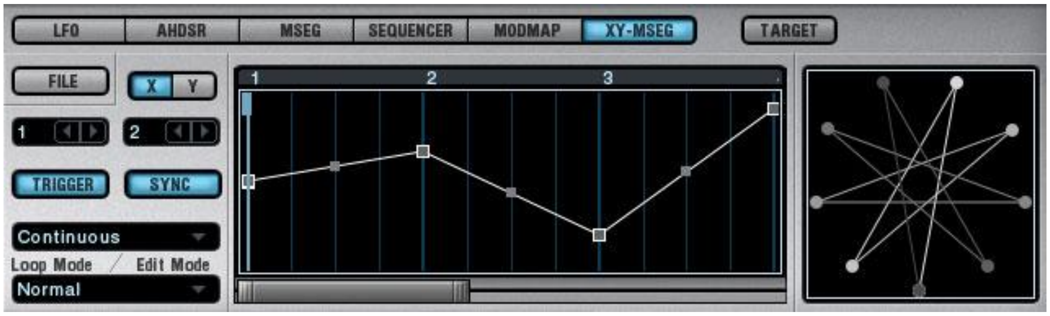
- A ***stepped ModMap*** ‘quantizes’ the input, mapping each input value to an output value defined by one of the steps.



XY-MSEG

An **XY MSEG** is not a separate type of modulator alongside Alchemy's AHDSRs, LFOs, and so on. Instead, it provides a special interface for manipulating Alchemy's regular MSEGs two at a time.

Specifically, an XY MSEG associates one MSEG with the X axis and another with the Y axis of a square, and it allows you to manipulate the levels of both MSEGs by dragging in the square. The MSEG XY editor also gives you a normal MSEG view, switchable between the X-axis MSEG and the Y-axis MSEG. In this view, you can change both the levels and the times of either MSEG.



When you change either times or levels in the XY MSEG editor, you're changing the associated (regular) MSEGs. By default, MSEG 1 is selected for the X-axis and MSEG 2 is selected for the Y-axis.

To modulate a pair of controls (such as Morph X and Morph Y) with an XY MSEG, **modulate one control with the X-axis MSEG** (by default, this is MSEG 1), and **modulate the other control with the Y-axis MSEG** (by default, this is MSEG 2). See the example 'MSEG-driven morphs' at the bottom of the Morph page for step-by-step instructions.

Using the XY MSEG controls

The XY MSEG **FILE** button opens a pop-up menu with the following commands:

- **Load** —Opens a dialog in which you can select a MSEG preset (*.mse) to load into the XY MSEG editor. If two presets in the same folder have identical names, but one name ends with 'X' and the other ends with 'Y' — for instance, 'Circle 1X.mse' and 'Circle 1Y.mse' — then selecting *either one* of these names loads the 'X' preset into the X-axis MSEG *and* the 'Y' preset into the Y-axis MSEG. If you select a name that is not part of an XY pair, then it will load just the selected file into the X- or Y-axis MSEG, depending on which axis button (see below) is activated.

- **Save** — Depending on which axis button (see below) is activated, saves the X- or Y-axis MSEG (but not both) to a preset file with a name of your choice.

- **Copy** and **Paste** — Copy places the X- or Y-axis MSEG information onto the clipboard; Paste retrieves the clipboard data and applies it to the X- or Y-axis MSEG.

- **Clear** —Sets the X- and Y-axis MSEGs to their default configurations.

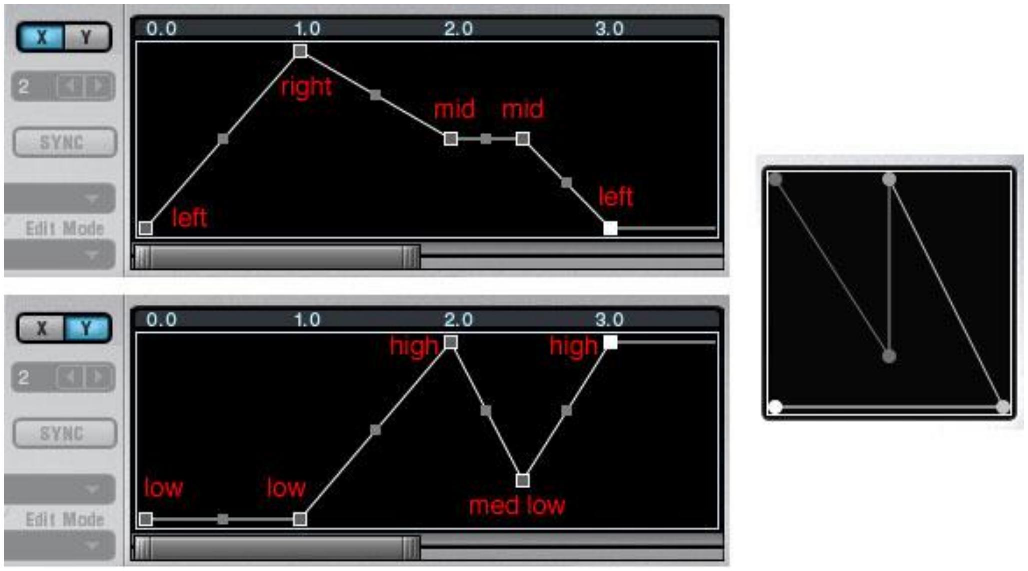
The **X** and **Y axis buttons** determine whether the X- or Y-axis MSEG is displayed and made available for editing in the breakpoint editor.

The **MSEG selection fields** determine which of Alchemy's MSEGs are associated with the X and Y axes of the XY MSEG square. The field on the left sets the X-axis MSEG, while the field on the right sets the Y-axis MSEG.

> ***Note: In order for the XY MSEG square to work properly, you should always assign two distinct MSEGs to the X and Y axes.***

The **TRIGGER**, **SYNC**, **Loop Mode** and **Edit Mode** controls work just like the controls of the same names on the normal MSEG control panel (see the MSEG page for details). The one difference in the XY MSEG editor is that each of these controls applies to both the X- and the Y-axis MSEGs.

Using the XY MSEG breakpoint editor and square



Working with the XY MSEG **breakpoint editor** is just like working with the breakpoint editor on the normal MSEG control panel (see the MSEG page for details) — except for one important difference: **the number and timings of breakpoints are kept synchronized between the X- and Y-axis MSEGs**. When you **create or delete points** — by right-clicking (control clicking) — you are increasing or decreasing the number of points in both MSEGs simultaneously. Similarly, when you **drag a point to a new time position**, you are changing the timing of a point in the X-axis MSEG and in the Y-axis MSEG. (On the other hand, the *levels* of the X- and Y-axis MSEG breakpoints are independent, so changing a level in one MSEG does not affect levels in the other MSEG.)

A quick illustration:

- Start with an initialized preset and open the XY MSEG editor.
- MSEG 1 is assigned to the X axis, and MSEG 2 is assigned the Y axis. Switch between the X and Y axis buttons to confirm that both of these MSEGs have a default shape consisting of three breakpoints.
- Click the X axis button and right-click (control click) in the breakpoint editor to add a fourth breakpoint to MSEG 1 at a time position of approximately 0.50 sec.
- Now switch to the Y axis view again and note that a fourth breakpoint has been added at the same time position in MSEG 2 as well.

If you have already edited one or both MSEGs using the MSEG control panel, then when you switch to the XY MSEG editor, the incoming MSEGs, which may arrive unsynchronized with one another, will immediately be synchronized — Alchemy performs this operation in such a way that the shape of both MSEGs is preserved, although the loop positions, loop mode, and trigger status of either MSEG may change.

While the XY MSEG editor's breakpoint envelope shows you the levels of either the X-axis MSEG or the Y-axis MSEG, the **XY square** shows you both at once. In the square, breakpoints are positioned such that their horizontal positions depict X values, while their vertical positions depict Y values.

A quick illustration:

- Start with an initialized preset and open the XY MSEG editor.
- Ensure that the X axis button is active, and then click the first (leftmost) breakpoint and drag it up and down. Note that the brightest breakpoint in the square moves to the left when you drag the X point down and to the right when you drag the Y point up. *The X value controls the left-to-right position in the square.*
- Now click the Y axis button to switch the view in the breakpoint envelope display, and then click the first (leftmost) breakpoint and drag it up and down again. This time, note that the brightest breakpoint in the square most to the bottom when you drag the Y point down and to the top when you drag the Y point up. *The Y value controls the bottom-to-top position in the square.*

While the XY square shows you both X and Y values, it doesn't show you the timing of each breakpoint; for that information, you'll need to refer to the breakpoint envelope display. However, the square does represent the 'chronological order' of the breakpoints by rendering them with different degrees of brightness, **from bright white for the first breakpoint through dark grey for the last**.

The XY square is not just a display, it's an editor. You can click a breakpoint in the square and drag it left/right to change its X value and up/down to change its Y value. To change the timing of a breakpoint, or to create or delete breakpoints, you should work in the breakpoint envelope display.

Tips and tricks (index)

Troubleshooting

If Alchemy is not behaving the way you expect, here are some things to try.

There is distortion — how do I get rid of it?

First, try turning down the main Volume control (in the Title bar), as the sound may be clipping when it enters your sequencer/host. If this does not solve the problem, then turn down all the Source Amp controls by equal amounts (a reduction of 6 dB or more may help). Sometimes by soloing different Sources and turning different elements on and off, you can isolate the problem in an individual source or element and resolve it by reducing the Source Amp or element Vol control. If distortion remains, it is likely to be caused by a deliberate distortion effect. Check to see if a source or main filter is set to the 'Tube', 'Mech', or 'Ring' type, or if a conventional filter type has a high Drive setting, or if the Effects rack contains a Distortion module.

The sound is too quiet — how do I make it louder?

Turn up the Master Volume and Master Amp controls (to their maximum levels if needed). If it is still too quiet, turn up Source Amp controls and individual element Vol controls. If you still don't have enough volume, try increasing the Drive setting of any conventional filters you are already using, or set an unused (source or main) filter to the 'Mech' type and boost its Resonance knob enough to raise the gain without causing noticeable distortion. You can also add a Compressor module to the Effects rack.

There is a 'click' at the start of each note — How do I get rid of it?

You have probably set the attack time of the Master Amp envelope to zero, which means the sound reaches full volume instantly. Try increasing the Attack time very slightly. A setting of 0.001 s, which Alchemy uses by default, often works well if you want a fast but click-free attack.

There are 'stuck' notes — how do I eliminate them?

If Alchemy continues to play notes after you release them, the likely culprit is the modulation routing of the Master Amp parameter. By default, AHDSR 1 modulates this parameter, acting as a ‘master amp envelope’; and whenever an AHDSR module is routed to the Master Amp parameter, the modulation depth is locked at 100%, which ensures that the amplitude will fall to zero at the end of the release stage. To resolve a ‘stuck’ notes issue, try the following:

- If you are using an MSEG as the ‘master amp envelope’, ensure that the modulation depth is set to exactly 100%.
- If you have not assigned either an AHDSR or an MSEG to modulate the Master Amp, add an AHDSR in this role. By setting an Attack time of 0 sec and a Sustain level of 100%, you can preserve the existing amplitude characteristics while eliminating the ‘stuck’ notes.
- Another possibility, external to Alchemy, is that your sequencer may contain MIDI data with a missing ‘Note Off’ message.

Alchemy’s CPU usage is too high — how do I reduce it?

First, if you are running your sequencer with less than 256 samples of latency, we recommend increasing it to 256 samples. Beyond that, there are a variety of strategies. The best solution depends on the details of your preset, so try the following, checking to see on one hand how much the CPU load is reduced and on the other hand how much the sound changes.

1. If using the Acoustic Reverb module with Quality set higher than 40%, reduce this setting to 40%. Or consider switching to the Camel Reverb effect.
2. Reduce polyphony. (Your needs for a particular preset may vary, but the following minimum voice counts were established for the factory presets: Keys, Organs 8; Pads, Brass, Guitars, Mallets, Strings, Synths, Vocals 6; Soundscapes, Drums, Arpeggiated 3; Bass, Loops, Sound Effects, Leads, Woodwinds 2. Of course some presets are monophonic by design.)
3. Turn STEREO off for all Sources which use the Additive element or the Source filters.
4. If using a HQ filter type, switch to the corresponding non-HQ type. (For example, switch from ‘LP2-SVF HQ’ to ‘LP2-SVF’; see the Filter page for details.)
5. Try switching off Source filters if their contribution to the preset is not significant.
6. If using the Additive element (in ADD or VA mode), use fewer oscillators. Listen to each Source as you reduce its NOsc control to ensure that you preserve the essential sound. Alchemy processes oscillators in blocks of four, so NOsc settings that are multiples of four (4, 8, 12, 16, …) give you the most oscillators for the least CPU cost.

7. The Additive element in ADD mode benefits from special optimizations, cutting CPU load in half when all of the following conditions are met:

    1. Oscillator wave is Basic > Sine.

    2. Symmetry is set to exactly 50% and not modulated.

    3. No modulation is applied to the Master Coarse and Fine Tune, Source Coarse and Fine Tune, and Additive element Pitch controls.

8. If using a Granular element with grain Density set above 4, reduce this setting to 4.

9. If the sound has a long release time, try reducing it; it may be more efficient to achieve similar effects with Delay and/or Reverb effects.

10. If using the Granular element with Stretch fixed at 100% and with no special granular manipulations, try switching the element to Sampler mode.

11. Use fewer effects.

12. Use fewer sources.

13. If in Xfade mode, use an analogous Morph mode instead. This makes a large difference for Additive and Spectral elements (although it may significantly change the sound of the preset).

A dialog box has appeared saying 'Please select a location and filename for supporting data files…' — what should I do?

This dialog box appears when your sequencer is saving a song that contains more than 1MB of additive or spectral analysis data. You should click 'OK'. In the following File Save dialog you should navigate to the folder in which your song is saved and enter a suitable name for the Alchemy preset file that will be saved there. This file will be referred to by the song so it is a good idea to keep it in the same folder.

The reason why it is necessary to save this data outside the song is because some sequencers do not allow more than 1MB of data to be stored inside a song by a plugin. Known examples of this are Logic and GarageBand. To find out if your sequencer has this limitation, please contact the sequencer manufacturer. If your sequencer does allow you to save more than 1MB of plugin data within a song, then you can increase this threshold by changing the value of the MaxChunkSize parameter within the AlchemyConfig.txt file that is located in your plugin folder. The MaxChunkSize parameter specifies

the threshold in bytes. Increasing this value when your sequencer does not support it will lead to a crash when saving songs.

There are no presets — why is that?

You should be able to browse the Factory presets, as well as any User presets you have saved, and any add-on banks of presets you have installed, via the Bank, Category, and Preset fields in Alchemy's Title bar. If the expected content is not displayed when you click these fields, then you may not have installed the Bank folders where Alchemy expects to find them. Locate these folders on your computer and move them to the Achemy data folder (Windows) or /Library/Application Support/Camel Audio/Alchemy/Presets (Mac); Alchemy will find them the next time it is loaded.

If you cannot find Alchemy's Factory bank anywhere on your computer, go to the Camel Audio web site at http://www.camelaudio.com, and log into your user account by clicking on the 'Log in / create user account' link at the top right-hand corner of the page and entering your email address and password. Then click on the 'Downloads' link in the Support Menu on the left of the page, and download Alchemy Presets (a .rar file that will be expanded by the installer) to your desktop. Then run the Add-On Installer, found in the Start menu under Programs->Camel Audio->Alchemy (Windows) or in /Applications/Alchemy (Mac).

Alchemy is asking for a 'keyfile' — what should I do?

The keyfile is your license to use Alchemy, and downloading it is a required step in the installation process. You can download the keyfile for your purchase (of Alchemy, or of an add-on bank for it) by logging in to your account at http://www.camelaudio.com and going to the Downloads page; see the Installation and requirements page of this manual for details.

If Alchemy or its add-on content is not already authorized, it looks for the keyfile on your desktop (and moves it automatically to the proper location). Please ensure you have placed your keyfile on the desktop, and then restart your sequencer. If you continue to see the keyfile message, please contact customer service for assistance.

Credits

Lead Programmer & Designer: Ben Gillett

Programmers: Rob Martino, John Proctor, Kelly Fitz, Magnus Jonsson, Jules Vleugels

Lead Sound Designers: Tim Conrardy, Colin ‘biomechanoid’ Fraser

Sound Designers: Arksun, Artvera, Beej, Antonio Blanca, Ian Boddy, Dangerous Bear, Richard Devine, Rory Dow, Torben Hansen, Junkie XL, Michael Kastrup, Christian Kjeldsen, Bryan ‘Xenos’ Lee, John ‘Skippy’ Lehmkuhl, Paul Nauert, Frank ‘Xenox’ Neumann, Pendle Poucher, Tasmodia

Sample Library: Artemis, Ian Boddy, Tim Conrardy, Rory Dow, Dunk, Claire Fitch, Colin ‘biomechanoid’ Fraser, Galbanum, Ben Hall, Nucleus SoundLab, Robert Rich, Brad Scherick, Scott Solida, Spe3d, Allen Strange, Neil Wakling

Graphic Interface: Bitplant

Manual: Paul Nauert, Paul Sellars

The design of Alchemy has been a collaborative effort. Many of the people listed above have had input into its design, varying from fundamentally shaping the nature of what Alchemy is, to suggesting small additional features.

Thanks to our customers, Magnus ‘Sonic Charge’ Lidström (for being a great friend when needed and all round genius), FXPansion (special thanks to Angus for all his advice over the years), Big Tick Audio Software (Rhino synthesizer), Lawrence Fritts (University of Iowa Electronic Music studios), Steve Holt (SfZed software), Richard Hotchkiss (Gvox music software), Dennis Lenz (Prodyon Robocoder software), Glen Olander (Green Oak Crystal synthesizer), Redmatica (the sample library was prepared with a custom version of KeyMap), all the other people who have helped us along the way and all our friends and family for putting up with us being glued to the screen for so long. Thank you all — we couldn’t have done it without you!

Manual History (10 January 09)

2 January 09 — added *Supported SFZ opcodes*

4 January 09 — added section bookmarks at the top of each longer page

4 January 09 — added Tips and tricks (index)

4 January 09 — added Manual History

8 January 09 — added details about warp markers and time-alignment to the description of morph modes on the Morph page

10 January 09 — added LFO shapes illustrated