

# MPowerSynth documentation



**MELDA** production  
*The only limit is your imagination*

[www.meldaproduction.com](http://www.meldaproduction.com)

[info@meldaproduction.com](mailto:info@meldaproduction.com)

MeldaProduction (c) 2016

# Installation

All MeldaProduction plugins are currently available for Windows and Mac OS X operating systems, both 32-bit and 64-bit versions. You can download all software directly from our website. Since the installation procedures for the two operating systems are quite different, we will cover each one separately.

The download files for the effects include all the effects plug-ins and MPowerSynth. During the installation process you can select which plug-ins or bundles to install. If you have not licensed all of the plugins in a bundle then you just need to activate each plugin separately.

If you have multiple user accounts on your computer, always install the software under your own account! If you install it under one account and run it under a different one, it may not have access to all the resources (presets for example) or may not even be able to start.

## Installation on Windows

All plugins are available for VST, VST3 and AAX interfaces. The installer automatically installs both the 32-bit and 64-bit versions of the plugins.

**Note: Always use 32-bit plugins in 32-bit hosts, or 64-bit plugins in 64-bit hosts. 64-bit plugins cannot work in 32-bit hosts even if the operating system is 64-bit. Conversely, never use 32-bit plugins in 64-bit hosts. Otherwise they would have to be 'bridged' and, in some hosts, can become highly unstable.**

You can select the destination VST plugins paths on your system. The installer will try to detect your path, however you should check that the correct path has been selected and change it if necessary. In all cases it is highly recommended to use the current standard paths to avoid any installation issues:

32-bit Windows:

C:\Program files\VstPlugins

64-bit Windows:

C:\Program files (x86)\VstPlugins *(for 32-bit plugins)*

C:\Program files\VstPlugins *(for 64-bit plugins)*

If your host provides both VST and VST3 interfaces, VST3 is usually preferable. If a plugin cannot be opened in your host, ensure the plugin file exists in your VST plugin path and that if your host is 32-bit, the plugin is also 32-bit, and vice versa. If you experience any issues, contact our support via [info@meldaproduction.com](mailto:info@meldaproduction.com)

## Installation on Mac OS X

All plugins are available for VST, VST3, AU and AAX interfaces. Installers create both 32-bit and 64-bit versions of the plugins.

If your host provides multiple plugin interface options, VST3 is usually preferable. If you experience any issues, contact our support via [info@meldaproduction.com](mailto:info@meldaproduction.com)

Most major hosts such as Cubase or Logic should work without problems. In some other hosts the keyboard input may be partly non-functional. In that case you need to use the virtual keyboard available for every text input field. You may also experience various minor graphical glitches, especially during resizing plugin windows. This unfortunately cannot be avoided since it is caused by disorder in Mac OS X.

## Uninstallation on Windows

The Uninstaller is available from the Start menu and Control panel, in the same way as for other applications. If you don't have any of these for any reason, go to Program files / MeldaProduction / MAudioPlugins and run setup.exe.

## Uninstallation on OSX

The Uninstaller is available from Applications / MeldaProduction / MAudioPlugins / setup.app.

## Performance precautions

In order to maximize performance of your computer and minimize CPU usage it is necessary to follow a few precautions. The most important thing is to keep your buffer sizes (latency) as high as possible. There is generally no reason to use latency under 256 samples for 44kHz sampling rates (hence 512 for 96kHz etc.). Increasing buffer sizes (hence also latency) highly decreases required CPU power. In rare cases increasing buffer sizes may actually increase CPU power, in which case you can assume your audio interface driver is malfunctioning.

You should also consider using only necessary features. Usually the most CPU demanding features are upsampling and modulation of certain parameters. You can reduce modulation CPU usage at the cost of lower audio quality in Settings/Settings/Modulator protection.

## Troubleshooting

The plugins are generally very stable, there are known problems however.

## GPU compatibility

The software uses hardware acceleration to move some of the processing (mainly GUI related) from your CPU (processor) to your GPU (graphics processing unit). It is highly recommended to use a new GPU, as it will provide higher performance improvements, and update your GPU drivers. Older GPUs are slower and may not even provide required features, so the software will have to perform all calculations in the main CPU. We also have had extremely bad experiences with GPUs from ATI and despite the fact that software is now probably bulletproof, it is recommended to use NVidia GPUs as there has not been a single case of a problem with them.

If you experience problems with your GPU (crashing, blank/dysfunctional GUI), and that you cannot disable the GPU acceleration from the plugin's Settings window itself, download this file:

<http://www.meldaproduct.com/download/GPU.zip>

And place the GPU.xml included in the zip into

Windows: C:\Users\{username}\AppData\Roaming\MeldaProduction

Mac OS X: ~/Library/Application support/MeldaProduction

## Memory limits of 32-bit platform

Most hosts are now 64-bit ready, however some of them are not or users willingly choose 32-bit edition, because the required plugins are not 64-bit ready yet. All our software is 64-bit ready. Please note that you must NOT use the 64-bit plugins in 32-bit hosts, even if you have a bridge. If you are stuck with a 32-bit host for any reason, note that there is a memory limit (about 1.5 GB), which you may not exceed. This can happen if you load too many samples or different plugins for example. In that case the host may crash. There is no other solution than to use a 64-bit host.

## Updating

You can use "Home/Check for updates" feature in any of the plugins. This will check online if there is a newer version available and open the download page if necessary.

To install a newer (or even older) version you simply need to download the newest installer and use it. There is no need to uninstall the previous version, the installer will do that if necessary. You also do not need to worry about your presets when using the installer. Of course, frequent backup of your work is recommended as usual.

## Using touch-screen displays

Touch screen displays are supported on Windows 8 and newer and the GUI has been tweaked to provide a good workflow. Up to 16 connections/fingers/inputs are supported. Any input device such as touch-screens, mouse, tablets are supported. These are the main gestures used by the plugins:

- Tap = left click
- Double tap = double click
- Tap & hold and quickly tap next to it with another finger = right click. Tap & hold is a classic right-click gesture, however that doesn't provide a good workflow, so came up with this method, which is much faster and does not collide with functionality of some elements.

## Purchasing and activation



You can purchase the plugin from our website or any reseller, however purchasing directly from our website is always the quickest and simplest option. The software is available online only, purchasing is automatic, easy and instant. After the purchase you will immediately receive a keyfile via email. If you do not receive an e-mail within a few minutes after your purchase, firstly check your spam folder and if the email is not present there, contact our support team using [info@meldaproduction.com](mailto:info@meldaproduction.com) so we can send you the licence again.

To activate the software simply **drag & drop the licence file onto the plugin**. Unfortunately some hosts (especially on Mac OS X) either do not allow drag & drop, or make it just too clumsy, so you can use Home/Activate in any of the plugins and follow the instructions. For more information about activation please check the [online video tutorial](#).

You are allowed to use the software on all your machines, but only you are allowed to operate the software. The licences are "to-person" as defined in the licence terms, therefore you can use the software on all your computers, but you are the only person allowed to operate them. MeldaProduction can provide a specialized licence for facilities such as schools with different licence terms.

## Quick start with your host

In most cases your host will be able to recognize the plugin and be able to open it the same way as it opens other plugins. If it doesn't, ensure you did installation properly as described above and let your host rescan the plugins.

### Cubase

Click on an empty slot (in mixer or in track inserts for example) and a menu with available plugins will be displayed. VST2 version is located in MeldaProduction subfolder. However VST3 version is recommended and is located in the correct folder along with Cubase's factory plugins. For example, dynamic processors are available from the "Dynamics" subfolder.

To route an audio to the plugin's **side-chain** (if it has one), you need to use the VST3 version. Enable the side-chain using the arrow button in the Cubase's plugin window title. Then you can route any set of tracks into the plugin's side-chain either by selecting the plugin as the track output or using sends.

To route **MIDI** to the plugin, simply create a new MIDI track and select the plugin as its output.

### Logic

Choose an empty insert slot on one of your audio tracks (or instrument tracks for example) and select the plugin from the popup menu. You will find it in the Audio Units / MeldaProduction folder.

To route an audio to the plugin's **side-chain** (if it has one), a side-chain source should be available in the top of the plugin's window, so simply select the source track you want to send to the plugin's side-chain.

To route **MIDI** to the plugin, you need to create a new Instrument track, click on the instrument slot and select the plugin from AU MIDI-controlled Effects / MeldaProduction. The plugin will receive MIDI from that track. Then route the audio you want to process with the plugin to this track.

## Studio One

Find the plugin in the Effects list and drag & drop it onto the track you would like to insert the plugin to.

To route an audio track to the plugin's **side-chain** (if it has one), first enable the side-chain using the "Side-chain" button in the Studio One's plugin window title. Then you can route any set of tracks into the plugin's side-chain from the mixer.

To route **MIDI** to the plugin, simply create a new MIDI track and select the plugin as its output.

## Digital performer

In the Mixing Board, find an empty slot in the track you would like to insert the plugin to. Click on the field and select the plugin from the effects list.

To route an audio track to the plugin's **side-chain** (if it has one), choose the track you want to send using Side-chain menu, which appears at the top of the DP's plugin window.

To route **MIDI** to the plugin, simply create a new MIDI track in the Track view and select the plugin as its output.

## Reaper

Click on an empty slot in the mixer and a window with available plugins will be displayed. Select the plugin you want to open by double clicking on it or using Ok button.

*It is highly recommended to select all MeldaProduction plugins in the plugin window the first time you open it, click using your right mouse button and enable "Save minimal undo states". This will disable the problematic Undo feature, which could cause glitches whenever you change certain parameters.*

To route an audio track to the plugin's **side-chain** (if it has one), click on I/O button of the side-chain source track in the mixer. Routing window will appear, there you click "Add new send" and select the track the plugin is on. In the created send slot select the channels (after the "=>" mark) for the send, in stereo configuration 3/4 for example. Note that this way the whole track receives the side-chain signal and all plugins with it. It is possible to send it to a single plugin only, but it is more complicated, please check the Reaper's documentation about that.

To route **MIDI** to the plugin, create a new MIDI track and do the same thing as with side-chain, except you don't need to change output channels.

## Live

In Session view, select the track you would like to insert the plugin to. At the left top of

Ableton Live's interface, click on the Plug-in Device Browser icon (third icon from the top). From the plug-ins list choose the plugin (from MeldaProduction folder), double click on it or drag & drop it into the track.

The X/Y grid usually doesn't provide any parameters of the plugin. This is because the plugins have too many of them, so you have to select them manually. Check Live's documentation for more information.

To route an audio to the plugin's **side-chain** (if it has one), select the track you want to send to the side-chain and in the 'Audio To' menu, choose the audio track that has the plugin on it. Then in the box just below that select the plugin from the menu.

NOTE: Live does NOT support any interface correctly, it doesn't use the reported buses properly, hence it doesn't work with surround capable plugins. Therefore you need to use VST version, which reports only stereo capabilities by default.

To route **MIDI** to the plugin, create a new MIDI track and in the 'MIDI to' menu, choose the audio track that has the plugin on it. Note that in Live only the first plug-in on any track can receive MIDI.

## ProTools

In the mixer click an empty slot to insert the plugin to and select the plugin from the tree. The plugin may be present multiple times, once for each channel configuration (mono->stereo etc.). As of now ProTools do not arrange them in the subfolders, which is a workflow dealbreaker, but we cannot do anything about it. The huge empty space on top of each plugin window, which occupies so much of the precious display area, is part of ProTools and every plugin window and again we cannot do anything about it. In some cases you may experience CPU overload messages, in which case please contact Avid for support. Note that ProTools 10 and newer is supported. RTAS compatibility for PT9 and older will never be added.

To route an audio to the plugin's **side-chain** (if it has one), open the plugin, click on the *No key input* button in the plugin title and select the bus you want the audio taken from. You might need to remember the bus number, unless your ProTools version supports bus renaming. ProTools doesn't support stereo (or surround) side-chains at all.

To route **MIDI** to the plugin, create a new MIDI track and in the mixer click the output field for that track and select the plugin, which should already be in the menu.

## GUI styles, editor modes and colors

MeldaProduction plugins provide a state of the art styling engine, which lets you change the appearance to your liking. The first time you run the plugins a style wizard will appear and let you choose the style and other settings. It may not be available in ProTools and other problematic hosts.

By default each plugin has a certain color scheme, which differs based on what kind of plugin is that. Also, sections of some plugins are colorized differently, again, based on what kind of section is that (this can be disabled in global settings). Despite you can change the colors anyhow you want, it is advantageous to keep the defaults as these are standardized and have predefined meaning, so just by looking at a plugin's color you can immediately say what kind of plugin and section is that. Same rules apply when designing active presets for easy screens.

This is the current set of colors:

Dynamics = orange

Equalization, filtering = green

Reverb, delay = brown/yellow

Modulation = blue

Distortion, limiting = red

Stereo = cyan/yellow

Time, pitch, unison... = purple/pink

Tools = grey

Special colors:

Synchronization = grey

Detection = blue/green

Side-chain = green

Effects = red

Advanced stuff = grey

# MeldaProduction MPowerSynth



## Overview

MPowerSynth is a deep synthesizer, featuring 3 oscillators, a noise generator, 2 filters and a modular effect pipeline. It also has 8 modulators, 8 multiparameters, an arpeggiator, a harmony generator, safety limiter and much more. The processing path is:

Voice1 = [Osc1] -> [Osc2] -> [Osc3] -> [Noise] -> [Filter1] -> [Filter2] -> Global ADSR ->  
Voice2 = [Osc1] -> [Osc2] -> [Osc3] -> [Noise] -> [Filter1] -> [Filter2] -> Global ADSR -> +  
-> [FX]

...

"[xxx]" marks optional items that may be enabled or disabled.

Each of the 3 oscillators features our adjustable shape technology and provides the best sounding engine on the market, which doesn't suffer from "digital not analog" problems. Oscillator 1 is a little different from the others - it has a unison feature, which generates up to 10 voices with slight pitch differences and generates a full wide sound. Oscillators 2 and 3 do not provide unison, instead these offer different methods of merging the output from the previous section with the oscillator signal. By default the method is simply mixed, but you can also use frequency modulation, ring modulation, convolution and more. Both oscillators 2 and 3 can have a dedicated ADSR envelope.

Noise generator contains several noise options, plus a low-pass and high-pass filters. It can also have a dedicated ADSR envelope. It is mixed with the output of the previous section.

There are 2 deep filters available. Each one has a dedicated extended ADSR driving frequency, resonance and other parameters. Unlike classic filters the frequency is specified in octaves;

that is a shift from the note frequency. So for example, if the note frequency is 100Hz and the frequency setting is +1 octave, then the central frequency of the filter is 200Hz, one octave higher than the note pitch.

There are more than 100 filter types available - classic LP/HP/BP/notch filters with slopes up to 120dB/octave, peak/shelf, harmonics, sub-X and other combined filters, format, comb/diffuser and polymorph filters. Most of them are pretty unique. The CPU consumption of different filters varies a lot; the most CPU-demanding ones are high-slope LP/HP/BP/notch filters, combined filters, diffusers and polymorph filters.

FX section presents the most advanced modular effect engine on the market, featuring more or less all our high-quality effects in a modular system. The engine is processing the output of all voices mixed together, so you don't need to worry about CPU that much anymore. Parameters of the effects loaded into the modular system cannot be automated, simply because you never know which effects that you will be using. However you can use modulators and multiparameters to control them, so if you need to automate any of them, you can just attach a multiparameter to that individual parameter and automate the multiparameter instead.



## Randomize button

Randomize button (with the text 'Random') generates random settings. Generally, randomization in plug-ins works by selecting random values for all parameters, but rarely achieves satisfactory results, as the more parameters that change the more likely one will cause an unwanted effect. Our plugins employ a smart randomization engine that learns which settings are suitable for randomization (using the existing presets) and so is much more likely to create successful changes.

In addition, there are some mouse modifiers that assist this process. The smart randomization engine is used by default if no modifier keys are held.

Holding **Ctrl** while clicking the button constrains the randomization engine so that parameters are only modified slightly rather than completely randomized. This is suitable to create small variations of existing interesting settings.

Holding **Alt** while clicking the button will force the engine to use full randomization, which sets random values for all reasonable automatable parameters. This can often result in "extreme" settings. Please note that some parameters cannot be randomized this way.

Hold **Shift** while clicking the button to undo the previous randomization.



## Presets button

Presets button shows a window with all available presets. A preset can be loaded from the preset window by double-clicking on it, using the arrow buttons or by using a combination of the arrow keys and Enter on your keyboard. You can also manage the directory structure, store new presets, replace existing ones etc. Presets are global, so a preset saved from one project, can easily be used in another.

Holding **Ctrl** while pressing the button loads an existing preset, selected at random.

Presets can be backed up by using either the Export button, or by saving the actual preset files, which are found in the following directories:

Windows: C:\Users\{username}\AppData\Roaming\MeldaProduction

Mac OS X: ~/Library/Application support/MeldaProduction

Exported preset files can be loaded into the plug-in's preset store using the Import button. Or the preset files themselves can be copied into the directories named above.

Files are named based on the name of the plugin in this format: "{pluginname}presets.xml", for example: MAutopanpresets.xml or MDynamicspresets.xml. If the directory cannot be found on your computer for some reason, you can just search for the particular file.

*The plugin also supports an online preset exchange. If the computer is connected to the internet, the plugin connects to our server once a week, submits your presets and downloads new ones, if available. This central preset library is maintained manually in order to validate submitted presets and remove generally unusable presets. It may take some time before any newly-submitted presets become available. This feature relies on each user, so we strongly advise that any submitted presets be named and organised in the same way as the factory presets, otherwise they will be removed.*

### **Left arrow button**

Left arrow button loads the previous preset.

### **Right arrow button**

Right arrow button loads the next preset.

### **Randomize button**

Randomize button loads a random preset.

### **Sound button**

Sound button generates a note for the synthesizer to play. One click produces a note-on, the next one is a note-off. This can be useful to audition the current settings without using a MIDI keyboard or any other MIDI source to control the instrument.

### **button**

This button resets the plugin state. You can use it to force the plugin to report latency to the host again and to avoid any audio problems.

For example, some plugins, having a look-ahead feature, report the size of the look-ahead delay as latency, but it is inconvenient to do that every time the look-ahead changes as it usually causes the playback to stop. After you tweak the latency to the correct value, just click this button to sync the track in time with the others, minimizing phasing artifacts caused by the look-ahead delay mixing with undelayed audio signals in your host. It may also be necessary to restart playback in your host.

Another example is if some malfunctioning plugin generates extremely high values for the input of this plugin. A potential filter may start generating very high values as well and as a result the playback will stop. You can just click this button to reset the plugin and the playback will start again.

### **Settings button**

Settings button shows a menu with additional settings of the plugin. Here is a brief description of the separate items.

**Activate** lets you activate the plugin if the drag & drop activation method does not work in

your host. In this case either click this button and browse to the licence file on your computer and select it. Or open the licence file in any text editor, copy its contents to the system clipboard and click this button. The plugin will then perform the activation using the data in the clipboard, if possible.

There are 4 groups of settings, each section has its own detailed help information: **GUI & Style** enables you to pick the GUI style for the plug-in and the main colours used for the background, the title bars of the windows and panels, the text and graphs area and the highlighting (used for enabled buttons, sliders, knobs etc).

**Advanced settings** configures several processing options for the plug-in.

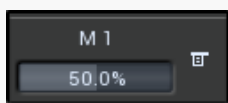
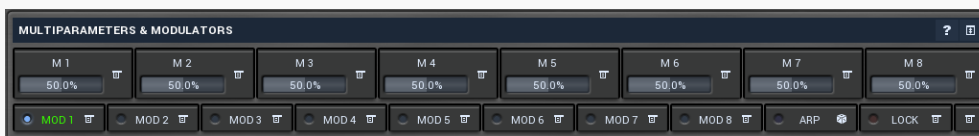
**Dry/wet affects** determines, for Multiband plug-ins, which multiband parameters are affected by the Global dry/wet control.

**Smart interpolation** adjusts the interpolation algorithm used when changing parameter values; the higher the setting the higher the audio quality and the lower the chance of zipping noise, but more CPU will be used.

## **button**

This button shows a menu with additional information about the plugin. You can check for updates, get easy access to support, MeldaProduction web page, video tutorials, Facebook/Twitter/YouTube channels and more.

# Multiparameters & Modulators



## **Multiparameter button**

Multiparameter button displays settings of the multiparameter. The multiparameter value can be adjusted by dragging it or by pressing Shift and clicking it to enter a new value from the virtual keyboard or from your computer keyboard.

Click on the button using your left mouse button to open the **Multiparameter** window where all the details of the multiparameter can be set. Click on it using your right mouse button or click on the **menu button** to the right to display an additional menu with learning capabilities - as described below.

## **button**

This button shows the **smart learn** menu. You can also use the right mouse button anywhere on the multiparameter button.

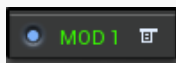


**Learn** activates the learning mode and displays "REC" on the button as a reminder, **Clear & learn** deletes all parameters currently associated with the multiparameter, then activates the learning mode as above. After that every parameter you touch will be associated to the multiparameter along with the range that the parameter was changed. Learning mode is ended by clicking the button again.

In smart learn mode the multiparameter does not operate but rather records your actions. You can still adjust every automatable parameter and use it normally. When you change a parameter, the plugin associates that parameter with the multiparameter and also records the range of values that you set.

*For example, to associate a frequency slider and make a multiparameter control it from 100Hz to 1KHz, just enable the smart learn mode, click the slider then move it from 100Hz to 1KHz (you can also edit the range later in the Multiparameter window too). Then disable the learning mode by clicking on the button.*

The multiparameter menu also contains items for reordering - this can be useful for creating active-presets. Note that this feature can cause problems when one multiparameter controls other multiparameters, these associations will not be preserved and will need to be reconstructed.



## Modulator button

Modulator button displays settings of the modulator. It also contains a checkbox, to the left, which you can use to enable or disable the modulator. Click on it using your right mouse button or use the **menu button** to display an additional menu with learning capabilities - as described below.

## button

This button shows the **smart learn** menu. You can also use the right mouse button anywhere on the modulator button.

**Learn** activates the learning mode and displays "REC" on the button as a reminder, **Clear & learn** deletes all parameters currently associated with the modulator, then activates the learning mode as above. After that every parameter you touch will be associated to the modulator along with the range that the parameter was changed. Learning mode is ended by clicking the button again.

In smart learn mode the modulator does not operate but rather records your actions. You can still adjust every automatable parameter and use it normally. When you change a parameter, the plugin associates that parameter with the modulator and also records the range of values that you set.

*For example, to associate a frequency slider and make a modulator control it from 100Hz to 1KHz, just enable the smart learn mode, click the slider then move it from 100Hz to 1KHz (you can also edit the range later in the modulator window too). Then disable the learning mode by clicking on the button.*



## button

This button displays additional menu containing features for modulator presets and randomization.

## **Randomize button**

Randomize button generates random arpeggiator settings.



## **Lock button**

Lock button displays settings of the parameter lock. Click on it using your left mouse button to open the Parameter Lock window, listing all those parameters that are currently locked. Click on it using your right mouse button or use the **menu button** to display the menu with learning capabilities - **Learn** activates the learning mode, **Clear & learn** deletes all associated parameters and then activates the learning mode. After that, every parameter you touch will be associated with the lock. Learning mode is then stopped by clicking the button again.



## **Tab selector**

Tab selector switches between subsections.

## **Randomize button**

Randomize button generates random settings for the tab.

## **Presets button**

Presets button chooses a random preset for the tab.

# Globals panel

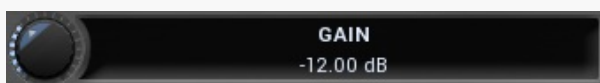


Globals panel contains some basic global settings, such as output volume or panorama

### **RANDOM NOTE PANORAMA**

## **Random note panorama**

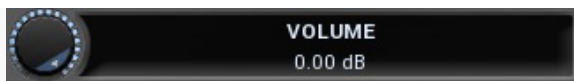
Random note panorama switch changes the behaviour of **Note panorama**. Normally that sets the panorama directly for each note, so you can for example use a modulator to control it. When this is enabled, it is merely a range and each note panorama will be randomly selected from the range  $-\text{panorama}.. \text{panorama}$ . Hence it also doesn't matter if current panorama is right or left.



## **Gain**

Gain defines the output gain. It is applied after all the generators and effects are performed.

Range: -48.00 dB to +48.00 dB, default 0.00 dB



## Volume

Volume defines the output volume adjustment, which is basically an alternative to gain with a different range. It is applied after all the generators and effects are performed.

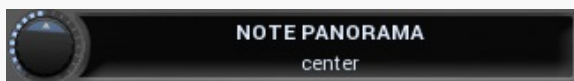
Range: silence to 0.00 dB, default 0.00 dB



## Panorama

Panorama defines the output panorama. It is applied after all the generators and effects are performed.

Range: 100% left to 100% right, default center



## Note panorama

Note panorama defines the panorama-per-note. It is applied after all the generators and effects are performed, but unlike global **Panorama**, this may be different for each note. The value is assigned at the moment the note is pressed and isn't changed later.

Range: 100% left to 100% right, default center



## Quality

Quality controls the ratio between audio quality and CPU requirements used by the signal generator. In almost all instruments the signal generators are the weakest point, because generating digital signals is extremely CPU demanding if it shouldn't be prone to aliasing and digital distortion. Therefore almost all synthesizers on the market provide inferior quality, sometimes masking it by adding noise and claiming it to be analog simulations. Our generators on the other hand are the finest on the market and in most cases Medium mode will provide nearly perfect sound quality. If you are using saw-tooth waves, frequency modulation and other features creating extremely harmonically rich signals, you may consider using High or even Highest quality.

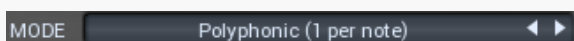
**Low** quality requires the least amount of CPU time, but you can experience aliasing (which may be used creatively of course). This is especially significant when using harmonically rich generator shapes, such as saw-tooth or square waves.

**Medium** activates an advanced generator, which requires slightly more CPU and memory resources, but minimizes aliasing. This mode is sufficient in most cases.

**High** improves the quality using more advanced interpolation.

**Highest** activates oversampling to at least 80kHz. This usually removes all aliasing and provides a true analog sound, in fact it provides a better than analog sound as it doesn't suffer from either analog or digital artifacts.

**Extreme** additionally uses extremely steep filters for the oversampling to minimize aliasing way below the limits of human hearing. This mode provides technically perfect generators. Its modifications such as **Extreme 16x** in addition increase the upsampling factor, which may be useful to remove distortion & aliasing when using transformations and additional mixing options such as frequency modulation. Please note that you can use the global upsampling as well, but this one is per-voice and may save CPU power as it is not upsampling the filters and FX section.



## Mode

Mode defines the way in which the synthesizer works.

**Polyphonic** mode is the default mode, which makes the synth create as many voices simultaneously as necessary.

**Polyphonic (single voice per note)** is similar to polyphonic, but ensures that only one voice will be created for each MIDI key. If sustain is pressed (or other conditions cause

multiple notes of the same key), the previous one will be terminated in the same way as when you release the key. This prevents voices stacking up when holding sustain pedal, however it may cause problems when harmonies are used, as pressing one key simulates pressing multiple keys.

**Monophonic** mode makes the synth play only one note at a time. When you hold one note and press another one the existing previous voice pitch is changed according to the new note, potentially with gliding. No attack stage or any kind of restart happens, so it sounds simply like the pitch has been changed.

**Monophonic (toggle)** mode is similar to the monophonic mode, however when you then release the new note while holding the previous one, the pitch of the voice is changed back to the note you are holding.

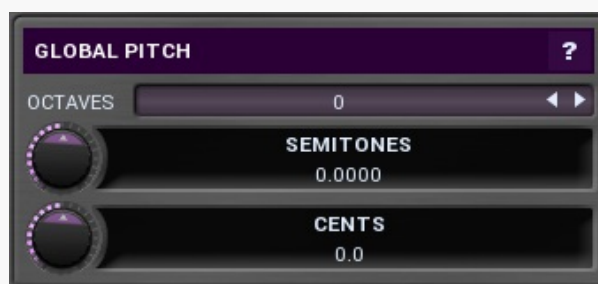
**Monophonic (attack), Monophonic (toggle + attack)** and **Monophonic (toggle + attack even off)** modes are similar to the 2 previous monophonic modes, however the voice is always switched back to the attack stage. This brings some focus to the new note, but no new voices are actually created. This mode smartly jumps in the ADSR envelope to the attack stage avoiding any abrupt changes in level and changes the pitch. The **even off** flag means that the attack is restarted even on note-offs. That means if you hold a long note, press a short one, then the attack is restarted not only when you press the short one, but also when you release it.

**Monophonic (brutal)** mode is similar to the previous modes, but the switch to the attack stage doesn't do any smoothing, so you may expect sharp clicking when the notes are restarting. This may be useful for short percussive sounds.

**Monophonic (restart)** mode and **Monophonic (toggle + restart)** mode are similar to the other monophonic modes, however instead of changing the pitch of the existing voices, the current voices are stopped in the same way as when you release them and new voices are created for the notes being pressed. This way the ADSR envelope is followed and you can get interesting overlaps of the release stages.

**Trigger** mode makes the synth completely ignore note release events and makes the notes be triggered on and off immediately providing a sort of staccato behaviour. This is especially useful for percussive sounds and in conjunction with another synth to form special attack sounds.

## Global pitch panel

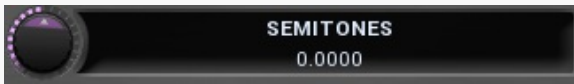


Global pitch panel lets you shift all generators and filters upwards or downwards.

**OCTAVES**  **Octaves**

Octaves defines the global pitch change in octaves.

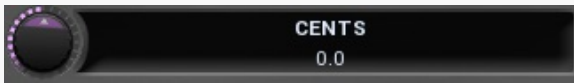
Range: -8 to +8, default 0



## Semitones

Semitones defines the global pitch change in semitones.

Range: -24.0000 to +24.0000, default 0.0000



## Cents

Cents defines the global pitch change (in cents of a semitone). The actual pitch change is the sum of these 3 control values.

Range: -100.0 to +100.0, default 0.0

# ADSR graph



ADSR graph controls the global ADSR envelope of the generated voices. Please note that additional oscillators, noise generators and filters can have their own ADSRs too. Our ADSR envelopes are much more sophisticated than classic attack-release-sustain-decay envelopes. Besides these common parameters they also let you control the curvature of each stage. Additionally, there are hold and delay sections, global smoothing and tremolo. You can even use the custom shape mode to define your own attack/release curves.

## **Random button**

Random button generates random settings using the existing presets.

## **Presets button**

Presets button displays a window where you can load and manage available presets. Hold **Ctrl** when clicking to load a random preset instead.

## **Left arrow button**

Left arrow button loads the previous preset.

## **Right arrow button**

Right arrow button loads the next preset.

## **button**

This button copies the settings onto the system clipboard.

## **button**

This button loads the settings from the system clipboard.

## Custom shape **Custom shape button**

Custom shape button enables custom shape mode, which lets you draw your own attack and release stages using the envelope system. Both stages are then automatically connected to form the resulting envelope.

## Immediate release **Immediate release button**

Immediate release button activates the immediate release mode in which case the note-off causes an immediate switch to the release stage. If this is disabled, the release stage does not occur until the whole attack/decay stage finishes.

## Sync **Sync button**

Sync button controls the ADSR tempo sync feature. By default this is disabled and means that all times are followed exactly, meaning that if **Attack** is say 100ms, then it will be 100ms indeed. Tempo sync lets the plugin adjust the times to ensure it will be always in sync with the host tempo. In this case 100ms may become say 125ms if the tempo is 120bpm, because 125ms is the length of a 16th note. This makes it extremely simple to convert any envelope to a tempo-synced one. The plugin always chooses the nearest longer note, in other words it always round up.

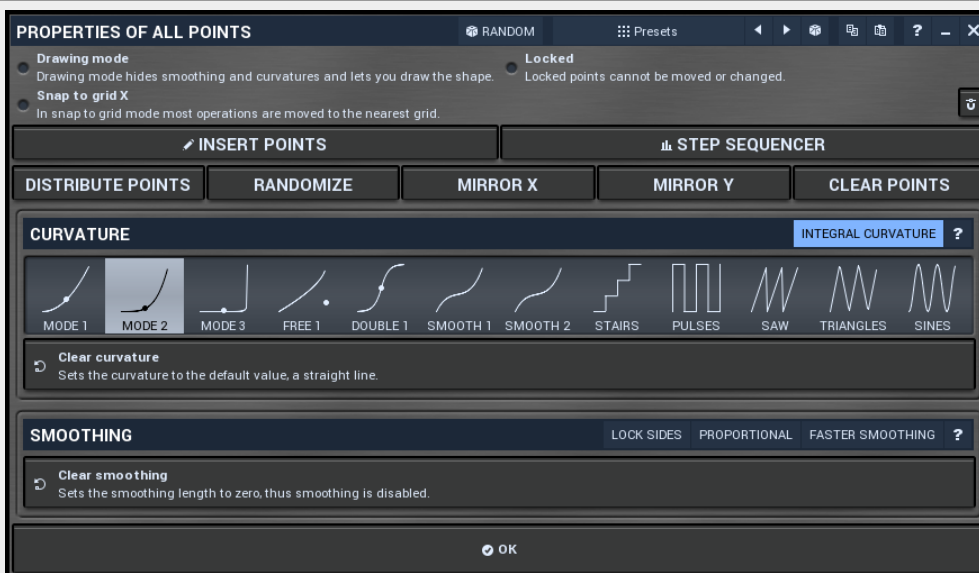
**Straight** and **Triplets** modes automatically find 'nice' values.

*For example, if a 16th note takes 100ms, the attack time is 550ms, and the sync mode is straight, then the plugin checks for 100ms, find out that it is too low, so it checks 8th note, being 200ms, still too low, then continues with quarter note, which takes 400ms, and still not enough, finally 800ms corresponding to a half note is the one, so the resulting time will be 800ms.* Triplet cases are more complex, but the principle is the same.

**1/16**, **1/8** and **1/4** modes choose the nearest higher multiply of the base note length.

For example, if a 16th note takes 100ms, the attack time is 550ms, and the sync mode is 1/16, the resulting time will be 600ms.

# Envelope graph menu





Envelope graph menu provides additional features which are used to edit the graph. Open the menu using right mouse button in the graph. Please note that if you select some points in the graph, or click on a point for example, the menu will be different and will cover only those features related to the selected set of points.

 **Random button**

Random button generates random settings using the existing presets.

 **Presets button**

Presets button displays a window where you can load and manage available presets. Hold **Ctrl** when clicking to load a random preset instead.

 **Left arrow button**

Left arrow button loads the previous preset.

 **Right arrow button**

Right arrow button loads the next preset.

 **Randomize button**


Randomize button loads a random preset.

 **button**


This button copies the settings onto the system clipboard.

 **button**

This button loads the settings from the system clipboard.

 **Insert point button**

Insert point button creates a point at mouse position.

 **Step sequencer**

**button**

Step sequencer button generates the envelope from step sequencer.

 **Distribute points button**

Distribute points button makes all points equally spaced.

 **Randomize button**

Randomize button slightly modifies the Y coordinates.

 **Mirror X button**

Mirror X button inverts the X coordinates of all points.

 **Mirror Y button**

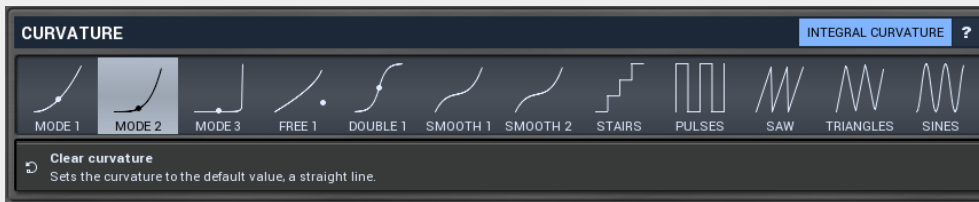
Mirror Y button inverts the Y coordinates of all points.

## CLEAR POINTS

### Clear points button

Clear points button deletes all points.

## Curvature

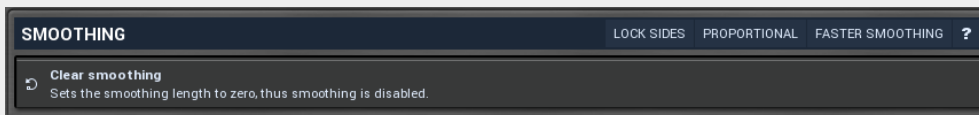


### INTEGRAL CURVATURE

### Integral curvature

Integral curvature makes the multi-curvature modes such as rectangles always have an integral number of items, e.g. 1, 2, 3, ... rectangles. If you disable this, it will be also possible to have for example 2.3 rectangles, which will however cause a discontinuity.

## Smoothing



### LOCK SIDES

### Lock sides

Lock sides makes the smoothing factor equal on both sides.

### PROPORTIONAL

### Proportional

Proportional makes the smoothing area size defined by the smaller side.

### FASTER SMOOTHING

### Faster smoothing

Faster smoothing enables slightly faster algorithm, which can however often cause unnecessary curving.



### Settings button

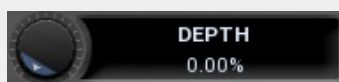
Settings button displays additional tremolo settings, containing tremolo behaviour and shape.

## Tremolo settings



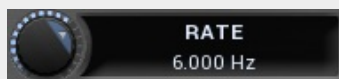


## Tremolo behaviour



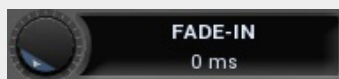
### Depth

Depth controls the amount of tremolo mixed in the sustain stage (or potentially before).



### Rate

Rate controls the tremolo rate and is relevant only if tempo sync is not used.



### Fade-in

Fade-in controls the length of the tremolo fade-in. It is especially useful when you want to use the random initial phase feature to avoid the initial discontinuity when the tremolo kicks in.



### sync

Tempo sync lets you synchronize the tremolo to the host's tempo.

## starts in decay stage

Tremolo starts in decay stage makes the tremolo start during the decay stage. By default this is disabled and the tremolo starts in the sustain stage. When it is enabled you will most likely have a longer decay and also a longer tremolo fade-in, so that the tremolo slowly comes in as the envelope is decaying.

## continues in release stage

Tremolo continues in release stage makes the tremolo continue with the tremolo during the release stage. By default this is disabled and the tremolo stops as soon as the release stage starts.

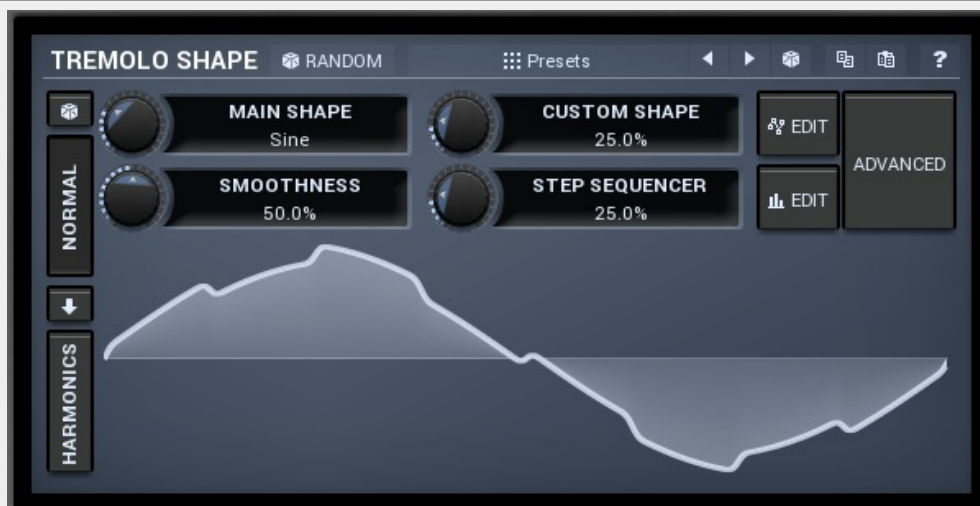
## initial phase

Random initial phase makes the tremolo start with a random phase. By default this is disabled and the tremolo starts always starts in the 0 phase, which ensures the tremolo always starts in the same way. However if you play multiple notes at once, the tremolo will be exactly the same, while you may want it to be different for each note and make it sound more 'human'. Enabling this option also activates a short **tremolo fade-in** to avoid initial discontinuity.

## sustain level

Follow sustain level makes the tremolo level based on sustain level. When this is disabled, the tremolo rarely reaches up to 100% level. However if the sustain level is say -20dB, then the tremolo actually cannot exceed 1% (which is -20dB), so it is clipped. It can however go upwards to 100%. This naturally changes the actual tremolo shape. If you want to avoid that and make sine really be a sine for example, enable this option, and in the case above the tremolo will really go up/down -20dB if set to 100%.

## Tremolo shape



## Random button

Random button generates random settings using the existing presets.

## Presets button

Presets button displays a window where you can load and manage available presets. Hold **Ctrl** when clicking to load a random preset instead.

### ◀ Left arrow button

Left arrow button loads the previous preset.

### ▶ Right arrow button

Right arrow button loads the next preset.

### 🎲 Randomize button

Randomize button loads a random preset.

### 📄 button

This button copies the settings onto the system clipboard.

### 📄 button

This button loads the settings from the system clipboard.

### 🎲 Randomize button

Randomize button generates random settings.

**NORMAL**

### Normal button

Normal button switches the generator into the normal mode, which lets you edit the shape of the oscillator. This is especially advantageous for low-frequency oscillators, where the shape matters even though it doesn't have any physical meaning.

### ⬇ Convert button

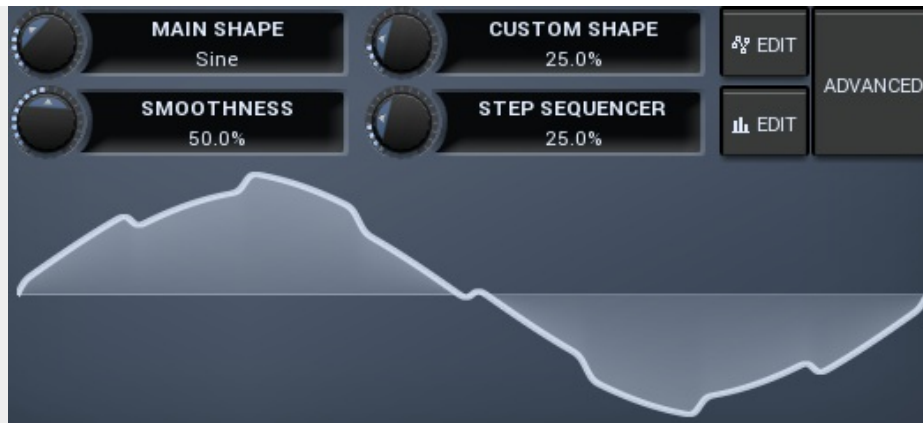
Convert button converts the current shape into harmonic-based representation. Please note that since the number of harmonics is limited, the result will not perfectly resemble the original shape.

**HARMONICS**

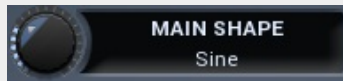
### Harmonics button

Harmonics button switches the generator into the harmonics mode, which lets you edit the levels and phases of individual harmonics. This is especially advantageous for high-frequency oscillators, hence sound generators.

## Signal generator in Normal mode

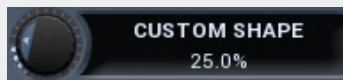


Signal generator in Normal mode works by generating the oscillator shape using a combination of several curves - a predefined set of standard curves, custom shape, step sequencer and custom sample. It also post-processes the shape using several filters including smoothing to custom transformations. This is especially useful when using the oscillator as an LFO (low-frequency-oscillator), where the harmonic contents does not really matter, but the shape does.



### **Main shape**

Main shape controls the main shape used by the signal generator. There are several predefined shapes, such as sine, triangle or rectangle, which you can choose from or even interpolate between using this control.



### **Custom shape**

Custom shape controls the amount of the custom shape that is blended into the main shape.



### **Edit button**

Edit button shows the custom shape editor.

## **Signal generator custom shape editor**



Signal generator custom shape editor controls the custom shape. You can edit virtually any shape that you can imagine and then blend it with the standard shapes, the step sequencer etc.

Presets

### Presets button

Presets button displays a window where you can load and manage available presets. Hold **Ctrl** when clicking to load a random preset instead.

### Left arrow button

Left arrow button loads the previous preset.

### Right arrow button

Right arrow button loads the next preset.

### Randomize button

Randomize button loads a random preset.

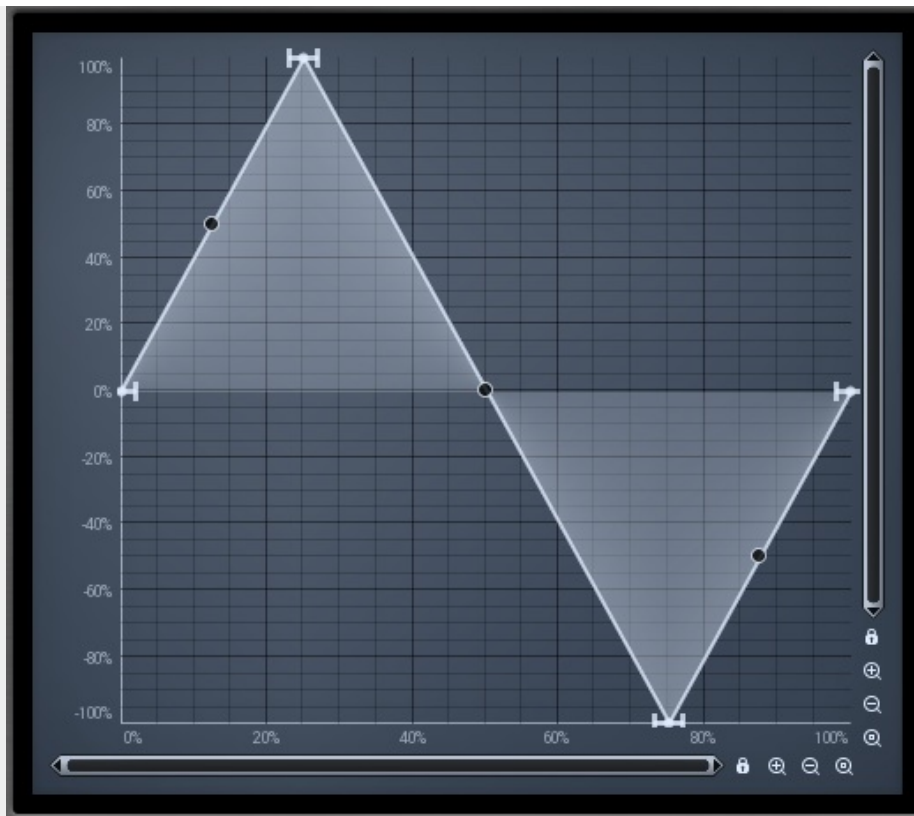
### button

This button copies the settings onto the system clipboard.

### button

This button loads the settings from the system clipboard.

## Graph editor



Graph editor lets you edit the envelope graph.

## Envelope graph

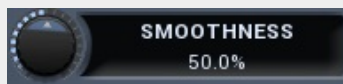
Envelope graph provides an extremely advanced way to edit any kind of shape that you can imagine. An envelope has a potentially unlimited number of points, connected by several types of curves with adjustable curvature (drag the dot in the middle of each arc) and the surroundings of each point can also be automatically smoothed using the smoothness (horizontal pull rod) control. You can also literally draw the shape in drawing mode (available via the main context menu).

- **Left mouse button** can be used to select points. If there is a *point*, you can move it (or the entire selection) by dragging it. If there is a *curvature circle*, you can set up its tension by dragging it. If there is a *line*, you can drag both edge points of it. If there is a *smoothing controller*, you can drag its size. Hold **Shift** to drag more precisely. Hold **Ctrl** to create a new point and to remove any points above or below.
- **Left mouse button double click** can be used to create a new point. If there is a *point*, it will be removed instead. If there is a *curvature circle*, zero tension will be set. If there is a *smoothing controller*, zero size will be set.
- **Right mouse button** shows a context menu relevant to the object under the cursor or to the entire selection. Hold **Ctrl** to create or remove any points above or below.
- **Middle mouse button** drag creates a new point and removes any points above or below. It is the same as holding Ctrl and dragging using left mouse button.
- **Mouse wheel** over a point modifies its smoothing controller. If no point



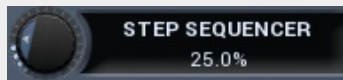
is selected, then all points are modified.

- **Ctrl+A** selects all points. **Delete** deletes all selected points.



## Smoothness

Smoothness controls the amount of smoothing. Many shapes, especially those produced by the step sequencer, have rough jagged edges, which may be advantageous, but when used to modulate certain parameters, the output may be clicking or causing other artifacts. Smoothness helps it by smoothing the whole signal shape out and removing these rough edges.



## Step sequencer

Step sequencer controls the amount of the step sequencer shape that is blended into the main shape (which has already been blended with the custom shape).



## Edit button

Edit button shows the step sequencer editor.

## Signal generator step sequencer editor



Signal generator step sequencer editor controls the step sequencer shape. You can have various numbers of steps each with a different value and shape. Note that for classic rectangular shapes the output can be very rough, hence it may be worth considering using **Smoothness** parameter to smooth out the resulting shape. This will use additional CPU power of

course, but that should be negligible unless you modulate any of the signal generator parameters.



### **Random button**

Random button generates random settings using the existing presets.



### **Presets button**

Presets button displays a window where you can load and manage available presets. Hold **Ctrl** when clicking to load a random preset instead.



### **Left arrow button**

Left arrow button loads the previous preset.



### **Right arrow button**

Right arrow button loads the next preset.



### **Randomize button**

Randomize button loads a random preset.



### **button**

This button copies the settings onto the system clipboard.



### **button**

This button loads the settings from the system clipboard.



### **Random values button**

Random values button generates random sequence of values, but keeps the shape of each step.



### **Random shapes button**

Random shapes button generates random sequence of shapes, but keeps the values of each step.

ADVANCED

### **Advanced button**

Advanced button displays an additional window with more advanced settings for post-processing the signal shape, such as harmonics or custom transformations.

## **Advanced settings**





### **Random button**

Random button generates random settings using the existing presets.

### **Presets button**

Presets button displays a window where you can load and manage available presets. Hold **Ctrl** when clicking to load a random preset instead.

### **Left arrow button**

Left arrow button loads the previous preset.

### **Right arrow button**

Right arrow button loads the next preset.

### **Randomize button**

Randomize button loads a random preset.

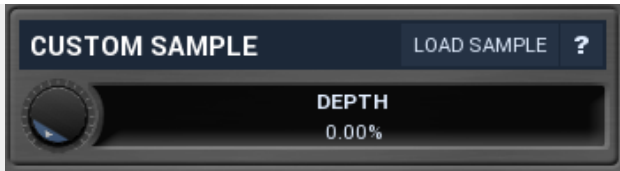
### **button**

This button copies the settings onto the system clipboard.

### **button**

This button loads the settings from the system clipboard.

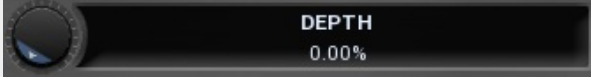
## **Custom sample panel**



Custom sample panel contains parameters of the custom sample that you can load and mix with the other sources. Do NOT confuse this with a sampler, the custom sample is taken as one period of the waveform. It can be used for creative effects and it can be used to import a custom waveform. The custom sample is then stored with limited precision within the settings, so the sample does not need to be kept on the system, but note that these settings may be quite large. To limit the space required by the settings, the sample is stored only if the depth is not 0%, meaning only if the sample is actually used.

**LOAD SAMPLE** **Load sample button**

Load sample button displays a file selection window, which lets you select the custom sample file.



**Depth**

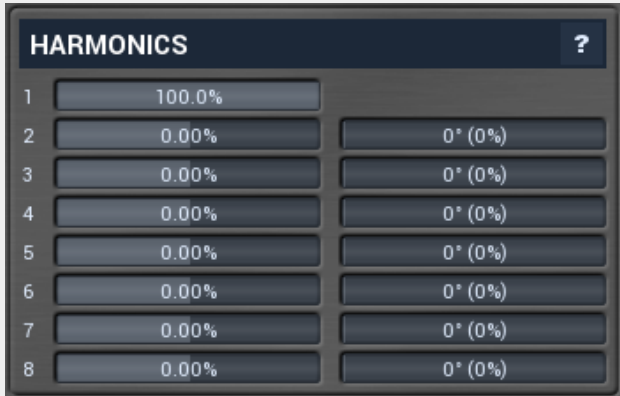
Depth controls the amount of custom sample mix. 0% means the sample is not used even if there actually is one loaded. 100% means the sample completely overrides the basic shape, custom shape, step sequencer... However, transformations are still performed on the sample.

**Shape panel**



Shape panel contains parameters performing various transformations on the signal shape. Please note that most transformation require a significant amount of CPU resources, so you should not automate or modulate the signal shape if you are using them.

**Harmonics panel**

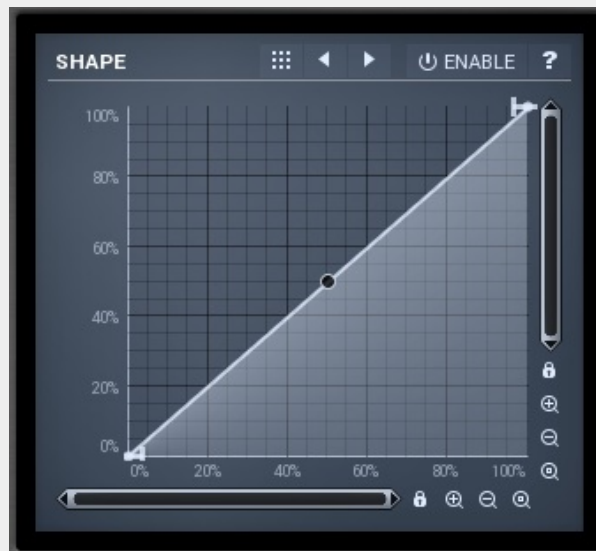


Harmonics panel lets you add separate harmonics of the original signal.

## Transformations



## Shape transformation graph



Shape transformation graph lets you perform arbitrary modification of the graph shape. Basically this graph lets you modify the shape "in time". The Y axis represents the position in the source signal related to the position in the target signal. The best way to check what it does is simply to try it.

### ▣ Presets button

Presets button displays a window where you can load and manage available presets. Hold **Ctrl** when clicking to load a random preset instead.

### ◀ Left arrow button

Left arrow button loads the previous preset.

### ▶ Right arrow button

Right arrow button loads the next preset.

## Amplitude transformation graph



Amplitude transformation graph lets you perform arbitrary modification of the graph amplitude. Basically this graph lets you modify the shape's level, vertical axis. The X axis represents the original values, the Y axis defines the resulting values. The best way to check what it does is simply to try it.

### ▣ Presets button

Presets button displays a window where you can load and manage available presets. Hold **Ctrl** when clicking to load a random preset instead.

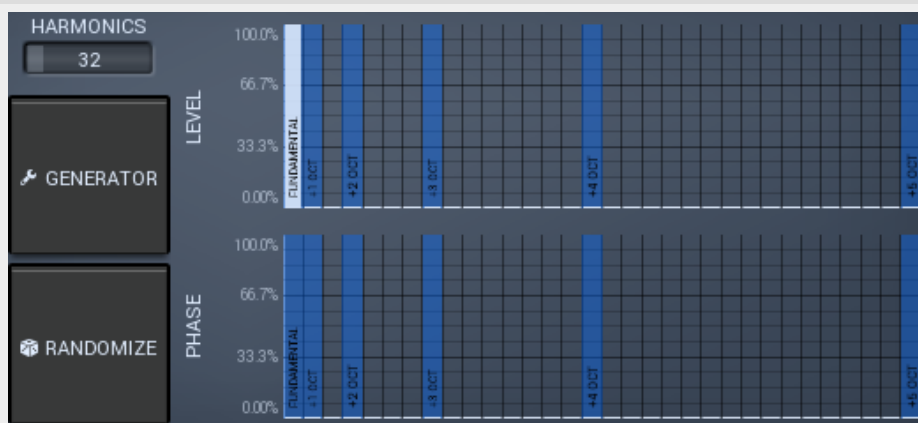
### ◀ Left arrow button

Left arrow button loads the previous preset.

### ▶ Right arrow button

Right arrow button loads the next preset.

## Signal generator in Harmonics mode

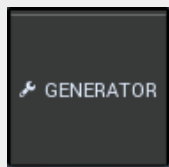


Signal generator in Harmonics mode works by generating the oscillator shape using individual harmonics. Essentially a harmonic is a sine wave. The first harmonic, known as the fundamental, fits once in the oscillator time period, hence it is the same as selecting sine wave in the **Normal mode**. The second harmonic fits twice, the third three times etc. In theory, any shape you create in normal mode can be converted into harmonics. However, this approach to signal generation needs an enormous number of harmonics, which is both inefficient to calculate and mostly hard to edit. Therefore, the harmonic mode can process up to

256 harmonics, which is enough for very complex spectrums, however it is still not enough to generate an accurate square wave for example. If your goal is to create basic shapes, it is better to use the normal mode.

It is nearly impossible to say how a particular curve will sound when used as a high-frequency oscillator in a synthesizer, just by looking at its shape. Harmonics mode, on the other hand, is directly related to human hearing and makes this process very simple. In general, the more harmonics you add, the richer the sound will be. The higher the harmonic, the higher the tone. Usually, one leaves the first harmonic enabled too, as this is the fundamental tone, however you may experiment with more dissonant sounds without it.

Editing harmonics can be time consuming unless you hear what you want, so a signal generator is also available. This great tool lets you generate a random spectrum by a single click. You can also open the **Generator<\reference> settings and edit its parameters, which basically control the audio properties in a more natural way - using parameters such as complexity, harmonicity etc.**



### button

This button shows a powerful harmonics generator, which can create unlimited number of various timbres and even analyze a sample and extract harmonics from it.

## Harmonics generator



Harmonics generator is a powerful tool, that can generate various harmonics-based timbres and even analyze a sample file and extract harmonics from it.



### Random button

Random button generates random settings using the existing presets.



### Presets button

Presets button displays a window where you can load and manage available presets. Hold **Ctrl** when clicking to load a random preset instead.

### ◀ Left arrow button

Left arrow button loads the previous preset.

### ▶ Right arrow button

Right arrow button loads the next preset.

### 🎲 Randomize button

Randomize button loads a random preset.

### 📄 button

This button copies the settings onto the system clipboard.

### 📄 button

This button loads the settings from the system clipboard.

## Generator panel



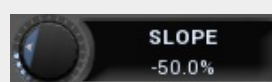
Generator panel contains parameters of the harmonics generator. By changing any of the parameters, the harmonics are changed, however only **Random seed** button changes the structure completely. The other parameters can be used to tweak the results.



### Harmonicity

Harmonicity controls the ratio between natural harmonics and those which sound disharmonic (despite the title "harmonics"). Assuming that the 1st harmonic is the fundamental, 2nd harmonic is 1 octave above, 4th is 2 octaves above, both can be considered very natural. 3rd harmonic is 1 octave and a 5th above the fundamental, and is still pretty harmonic, but less than the octaves. 5th harmonic is 2 octaves and a major 3rd above the fundamental. Such a tone may sound very disharmonic, in minor scales for example. Higher harmonics are often very disharmonic and produce typical ringing timbres.

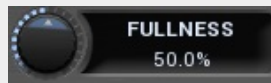
When harmonicity parameter is set to 100%, only octaves are allowed. By lowering the value more and more disharmonics are created and with 0% all frequencies are allowed. For values below 0% disharmonics are preferred, hence you can expect more ringing timbres.



### Slope

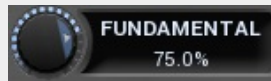


Slope defines the amount of higher harmonics compared to lower ones. When 0%, the higher harmonics have the same levels as lower ones. Typically you use values below 0%, which attenuates the higher harmonics making the resulting sound darker. Similarly values above 0% make the sound brighter.



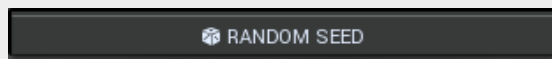
## Fullness

Fullness controls the number of generated harmonics. With values around 0% the resulting timbres will contain only a few harmonics making the sound clear. Higher values increase number of harmonics making the timbre rich.



## Fundamental

Fundamental controls the minimum level of the fundamental (the 1st harmonic). Most sounds have a very strong fundamental as it carries the pitch.



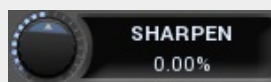
## Random seed button

Random seed button generates a new series of harmonics. Pressing this button will create a whole new timbre.

## Post-processor panel

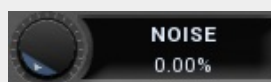


Post-processor panel contains parameters of the harmonics post-processor. The generator and sample analyzer first create a series of harmonics, the timbre. These harmonics are mixed depending on the **Sample ratio** parameter. After that the post-processor is engaged, which can further transform the harmonics in several ways.



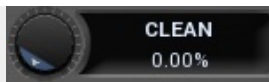
## Sharpen

Sharpen is a sort of soft compression/expanding unit. Values below 0% decrease the level of quiet harmonics, while values above 0% increase their level.



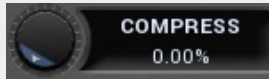
## Noise

Noise defines amount of noise added to the timbre. Noise can make the results dirty providing much richer timbres.



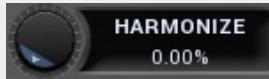
## Clean

Clean controls the threshold of a gate. It basically attenuates or removes harmonics below this level making the output cleaner.



## Compress

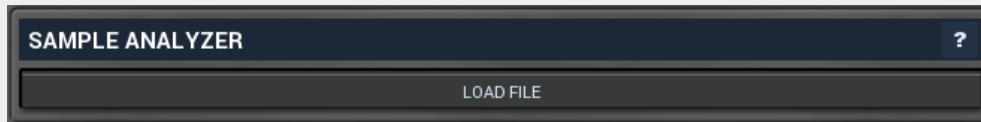
Compress reduces the dynamic range of the harmonics by increasing levels of the quiet ones, but keeping the levels of the loud ones.



## Harmonize

Harmonize creates additional higher harmonics from existing ones. This is especially useful to transform rich dirty disharmonic timbres into similarly rich but more harmonic timbres.

## Sample analyzer panel



Sample analyzer panel contains parameters of the sample analyzer. If there is no sample loaded, the sample analyzer is turned off. The analyzer takes the selected sample and a position within it, analyses one period of the signal waveform and produces the output set of harmonics. You can then combine these harmonics with the output of the generator using **Sample ratio** parameter.

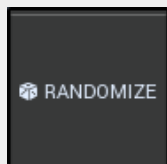
The sample itself is not stored with the plugin settings. Instead the path to the target sample file is stored along with the analyzed harmonics. If the sample file is not available, you cannot modify the analysis parameters and the last analyzed harmonics are used. This means that you actually don't need to have the sample file available on the computer on which you are using the settings.



**Load**

### file button

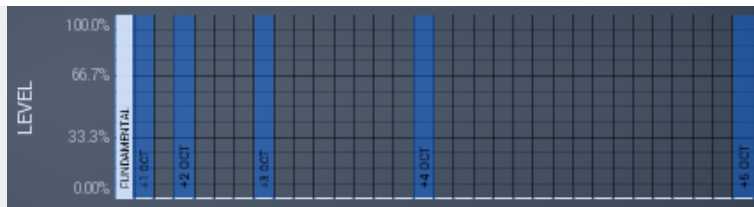
Load file button lets you select a sample file to analyzer.



### Randomize button

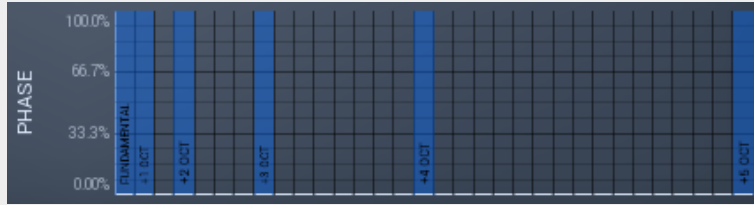
Randomize button selects random parameters for the harmonics generator, so you can use it to get a random sound character instantly. Hold **Ctrl** to slightly modify existing generator settings instead of completely changing them.





## Magnitudes graph

Magnitudes graph contains the levels of the individual harmonics. The highlighted bars are octaves, thus the 1st, 2nd, 4th, 8th harmonic etc.



## Phases graph

Phases graph contains the phases of the individual harmonics. The highlighted bars are octaves, thus the 1st, 2nd, 4th, 8th harmonic etc.

# Osc 1 panel



Osc 1 panel contains settings for the main oscillator. This first oscillator is different from the other oscillators as it doesn't offer any combination mode, but it has unison feature instead. In most cases you will leave this oscillator enabled. For noise-based settings you may want to disable it though.

### **Random button**

Random button generates random settings using the existing presets.

### **Presets button**

Presets button displays a window where you can load and manage available presets. Hold **Ctrl** when clicking to load a random preset instead.

## ◀ Left arrow button

Left arrow button loads the previous preset.

## ▶ Right arrow button

Right arrow button loads the next preset.

## 🎲 Randomize button

Randomize button loads a random preset.

## 📄 button

This button copies the settings onto the system clipboard.

## 📄 button

This button loads the settings from the system clipboard.

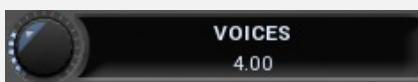
# Unison panel



Unison panel contains parameters of the unison generator, which essentially creates multiple clones for each voice making the sound rich and wide.

## HIGH DETUNE High detune

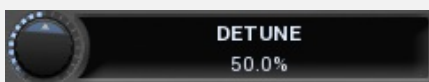
High detune extends the detuning range significantly making it an interesting tool for creative effects.



### Voices

Voices controls the number of voices of the unison generator. The number may be fractional in which case the last voice is not "fully generated".

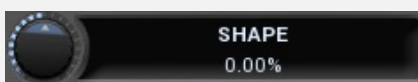
Range: 1.00 to 10.00, default 4.00



### Detune

Detune defines the amount of detuning for each unison clone.

Range: 0.00% to 100.0%, default 50.0%

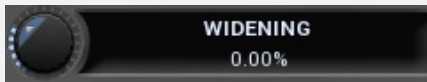


### Shape

Shape controls the shape of the frequencies for each unison voice. At 0% the frequencies will be equally displaced around the base pitch. At -100% most of the frequencies will be placed closely around the base pitch. Conversely, at +100%, they will tend towards the extremes and be as far away from the base pitch as possible,

the limit being defined by the **Detune** parameter.

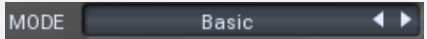
Range: -100.0% to 100.0%, default 0.00%



## Widening

Widening controls the stereo widening for the unison clones.

Range: Mono to 200.0%, default 0.00%



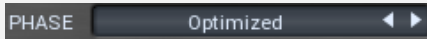
## Mode

Mode controls the way the unison voices cover the frequencies around the base pitch, hence the actual unison sound.

**Basic** is the original mode in which the voices are surrounding the base pitch, but the actual pitch may not actually be created unless the number of voices is at the maximum.

**Advanced** covers the frequencies around the base pitch up to the detune level, and the dispersion is controlled by the **Shape** parameter.

**No base pitch** is similar to Advanced mode, but similarly to the Basic mode it does not generate the actual base pitch.



## Phase

Phase controls the initial phase for the unison voices. The voices share the same signal shape and are very similar in frequency, so they interact quite a bit when their phases are getting close to each other. This causes natural phasing. This mode controls at which phase the voices start, which directly defines the character of the phasing, mainly in the beginning of the note.

**Optimized** is the default value, which optimizes the phasing to the minimum in most cases.

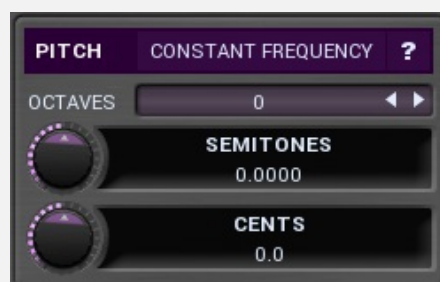
**Optimized wide** is similar but provides the maximum stereo width. Please note that this may cause mono-compatibility problems.

**Random** provides random phases every time. This will result in a slightly different character for each note.

**Random wide** is similar but provides the maximum stereo width. Please note that this may cause mono-compatibility problems.

**Zero** lets all voices start at the initial phase, therefore you may expect maximum phasing to occur at the beginning and slowly to disappear. Please note that this mode may cause the unison effect to disappear for minimum detune.

## Pitch panel



Pitch panel lets you shift the main generator upwards or downwards.

**CONSTANT FREQUENCY**

## Constant frequency

Constant frequency makes the main oscillator ignore the note pitch and behave as if an A4 (440Hz) note is received each time.

OCTAVES

0

### Octaves

Octaves defines the main oscillator pitch change (in octaves).

Range: -8 to +8, default 0

SEMITONES

0.0000

### Semitones

Semitones defines the main oscillator pitch change (in semitones).

Range: -24.0000 to +24.0000, default 0.0000

CENTS

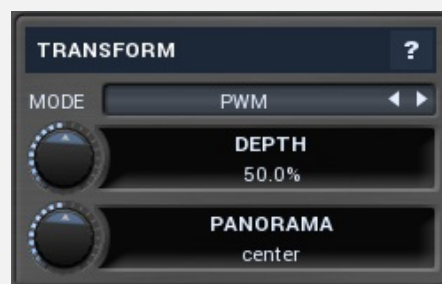
0.0

### Cents

Cents defines the main oscillator pitch change (in cents of a semitone). The actual pitch change is the sum of these 3 control values.

Range: -100.0 to +100.0, default 0.0

## Transform panel



Transform panel contains some additional oscillator transformation features.

MODE

PWM

### Mode

Mode controls the type of transformation used on the oscillator waveform. Mode controls the type of transformation used on the oscillator waveform. The oscillators themselves are pre-processed in a very complex manner to provide maximum audio quality. However that means that every time you change something in the oscillator editor, lots of calculations need to be performed, which may take quite some CPU. The transformations however are performed without pre-processing, so they may provide lower audio quality, but they are much less CPU intensive and often provide actually higher quality when modulated, because these are always sample accurate.

It is therefore recommended to use the oscillator editor for static sounds and then use transformations for modulation. Another reason to use the transformation is the actual harmonic properties of the generated sound - any oscillator (algorithmic or wavetable) produces only a fundamental frequency and its harmonics, the actual wave shape only controls the levels and phases of these frequencies. This means that if the fundamental pitch is say 100Hz, then the generated sound contains only 100Hz, 200Hz, 300Hz etc. It can never produce 150Hz for example. And that's when the

transformations comes in handy again. When you transform the waveform in any way the result is just another waveform, so it still cannot produce other frequencies. However if you change the transformation depth in real-time, say using modulators, the output could produce just about any frequencies, all depending on the actual transformation and the way it is modulated.

**Disabled** disables any transformation.

**PWM** implements several pulse-width modulation algorithms, all affecting the width of the pulse or bending its shape in time. PWM tends to produce frequencies around each harmonic when modulated. The quicker the PWM depth is changed, the further these frequencies are from the main harmonics and the sound gets richer until a point where it is completely disharmonic.

**Sync** essentially duplicates the periods of the waveform (not necessarily a multiple). The higher the depth, the higher the dominant harmonic, so moving the depth in a way controls the pitch without losing the sense of the actual fundamental pitch. Sync waveform has abrupt edges in most cases. These generate lots of higher harmonic content, sometimes even reaching zipper noise. To diminish this and to get a cleaner sound, use the windowed versions, which smooth out the edges.

**ASync** transformations are similar to Sync, however the waveforms are bent, so the typical sync screaming is less resonant and more inharmonic.

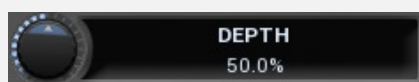
**Bending and mirroring** alter the waveform quite smoothly, so these produce less rich tones and the effects may remind you of phasing. Discontinuities may occur when modulated quickly, so you can exploit that to get a richer sound if needed.

**Reversing** merges the waveform with a reversed replica of itself. This may often cause abrupt edges producing a sort of a modulated square wave. To get the most out of it, the original waveform should not be (anti)symmetric - when you look at a sine wave for example, the right half is just an inverted left half, so reversing it isn't such a miraculous action. This is true for most basic waves including saw and rectangle for example. Therefore reversing is most interesting when used with some more complex wave-shapes produced by harmonic mode or the step sequencer etc.

**Invert, Maximize, Zero under, Power and Quantize** are manipulating the amplitude only and due to the discontinuities these produce fairly rich sounds, lo-fi in a way. Please note the processor may produce different results for different quality modes, because it can generate specific waveforms for different notes to improve quality, but since the amplitude will be different, the results of the transformation will be different as well.

**AM** provides amplitude modulation with itself or a reversed version of itself. As such it provides minimum non-harmonic frequencies.

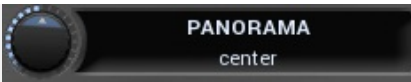
**Recursive** transformations are the most complex and basically transform the waveform by projecting it onto itself. The output generally depends on the input waveform signal and you might say that the more complex the initial waveform is, the more complex the output waveform will be. Recursive transformations may produce extremely full spectra and may mix the properties and richness of other transformations combined. As such, simple waveforms such as sine and similar usually produce the best results. Modulating a harmonically complex waveform can easily end up with white noise.



## Depth

Depth controls the amount/type of the selected transformation and could be used for modulation. Its meaning depends on the selected transformation.

Range: 0.00% to 100.0%, default 50.0%



## Panorama

Panorama defines the oscillator panorama transformation.  
Range: 100% left to 100% right, default center

## Signal editor



Signal editor controls the oscillator shape. You can use either oscillator shapes or harmonics; the latter are preferred in this case as they directly control the harmonic structure of the generated sounds. Signal-generator is an incredibly versatile generator of low & high frequency signals. It offers 2 distinct modes - Normal and Harmonics.

**Normal mode** is appropriate for low-frequency oscillators, where the graphical shape is relevant and is used to drive some form of modulation. For example, a tremolo uses this modulation to change the actual signal level in time. Frequencies for such oscillators usually do not exceed 20Hz as this is a sort of limit above which the frequencies become audible.

**Harmonics** mode is designed for high-frequency oscillators, where the actual shape is not as important as the harmonic content of the resulting signal, hence it is especially useful for actual audio signals. Please note that since a shape can contain more harmonics than those available from the harmonic generator, the results may not be exactly the same. As an example, a rectangular wave in normal mode may sound fuller than when converted to the harmonic mode.

Use the arrow-down button to switch from normal mode to harmonics mode or click the **Normal** and **Harmonics** buttons

## Normal mode

The generator first uses a set of predefined signal shapes (sine, triangle, rectangle...), which you can select directly by right-clicking on the editor and choosing the requested shape from the menu. This menu also provides a link to the modulator shapes preset manager, normalization and randomization. You can also use the **Main shape** parameter, which generates a combination of adjacent signals to provide a nearly inexhaustible number of basic shapes.

The engine then combines the predefined shape with a **Custom shape**, which may



be anything you can draw using the advanced envelope engine, depending on the level set by the **Custom shape** control. Use the **Edit** button to edit the custom shape.

You can also combine those results with a fully featured step sequencer, with variable number of steps and several shapes for each of them, depending on the level set by the **Step sequencer** control. Use the lower **Edit** button to edit the step sequence.

Those results may be mixed with a custom sample, which is available from the advanced settings, accessed by clicking the **Advanced** button.

**Smoothness** softens any abrupt edges, generated by the step sequencer for example.

Finally there are **Advanced** features providing more complex transformations, adding harmonics etc. or you can click the **Randomize** button in the top-left corner to generate a random, but reasonable, modulator shape.

## Harmonics mode

Harmonics mode represents the signal as a series of harmonics (that is, multiples of the base frequency). For example, when your oscillator has a frequency of 2Hz (set in the panel), then the harmonics are 2Hz, 4Hz, 6Hz, 8Hz etc. In theory, any signal can be created by mixing a potentially infinite number of these harmonics.

The harmonics mode lets you control the levels and phases of each harmonic. The top graph controls the levels of individual harmonics, while the bottom one controls their phases. Use the left-mouse button to change the values in each graph, the right-mouse button sets the default for the harmonics - 0% level and 0% phase. In both graphs the harmonics of power 2 (that is octaves) are highlighted. Other harmonics may actually sound disharmonic, despite their names.

*For example, if you reset all harmonics to the defaults and increase only the first one, you will get a simple sine wave. By adding further harmonics you make the output signal more complex.*

**Harmonics** controls the number of generated harmonics. The higher the number is, the richer the output signal is (unless the levels are 0% of course). This is useful to make the sound cleaner. For example, if you transform a saw-tooth wave to harmonics, it would not sound like a typical saw-tooth wave anymore, but more like a low-passed version of one. The more harmonics you use, the closer you get to the original saw-tooth wave.

**Generator** is a powerful tool for generating the harmonics, which are otherwise rather clumsy to edit. The generator provides several parameters based upon which it creates the entire series of harmonic levels and phases. These parameters are usually easier to understand than the harmonics themselves. Part of the generator is the randomizer available via the **Random seed** button, which smartly generates random settings for the generator. This makes the process of getting new sounds as simple as possible.

# Signal generation fundamentals

The signal generator produces a periodic signal with specified wave shape. This means that the signal is repeating over and over again. As a result it can only contain multiples of the fundamental frequency. For example, if the generator is producing 100Hz signal, then it can contain 100Hz (fundamental or 1st harmonic), 200Hz (2nd harmonic), 300Hz (3rd harmonic), 400Hz (4th harmonic) etc. However, it can never produce 110Hz. You can then control the level of each harmonic and their relative phases. It does not matter whether you use the normal mode using oscillator shapes, or harmonics mode where you can control the harmonics directly. If both modes result in the same wave shape (such as sine wave vs. 1st harmonic only), then the result is exactly the same.

Sine wave is the simplest of all as it contains the fundamental frequency only. The "sharper" the signal shape is, the more harmonics it contains. The biggest source of higher harmonics is a "discontinuity", which you can see in both rectangle and saw waves. In theory, these signals have an infinite number of harmonics. However since our hearing is highly limited to less than 20kHz, the number of harmonics which are relevant is actually pretty small. If you generate a 50Hz signal, which is very low, and assuming that you have extremely good ears and you actually hear 20kHz, then the number of harmonics audible for you is  $20000 / 50 = 400$ .

## What happens above 20kHz?

Consider the example above again, what happens with harmonics above 400? These either stay there and simply are not audible, disappear if anti-aliasing is used, or get aliased back under 20kHz in which case you get the typical digital dirt.

When you convert a rectangle wave to harmonics mode, only the first 256 harmonics are used, so it basically works like an infinitely steep low-pass filter. What is the limit then?  $50 \text{ Hz} * 256 = 12.8\text{kHz}$ . The harmonic mode will not produce anything above this limit if you are generating a 50Hz signal. Most people do not hear anything above 15kHz, so this is usually enough, but if not, you may need to use the normal mode where you get the "infinite" number of harmonics.

## What you see is not always what you get!

Say you want a rectangle wave and play a 440Hz tone(A4). You would expect the output signal to be a really quick rectangle wave, right? Wrong! If you would do that, and actually most synthesizers on the market do that, you would get the infinite number of harmonics. And, since you are working in say 48kHz sampling rate, the maximum frequency that can actually exist in your signal is 24kHz. So everything above it would get aliased below 24kHz, and there would be a lot of aliased dirt.

The "good" synthesizers perform a so-called anti-aliasing. There are several methods, most of them require quite a lot of CPU or have other limitations. The goal is to remove all frequencies above the 24kHz in our case or in reality, it is more about removing all aliased frequencies above 20kHz - this means, that we do not care about frequencies above 20kHz, because we do not hear them anyway. But we will keep it simple. Let's say we remove everything above 20kHz. You already know that the



rectangle wave can be created using an infinite number of harmonics or sine waves. We removed everything above the 45th harmonic (20000 / 440) so our rectangle wave is trying to be formed using just 45 harmonics, so it will not really look like a rectangle wave.

After some additional filtering (like DC removal), the rectangle wave may look completely different than a true rectangle wave, yet it would sound the same! Does it matter? Not really. You simply edit the shape as a rectangle wave and let the synthesizer do the ugly stuff for you. But do not check the output, because it may be very different than what you would expect ;).

## How can I generate non-harmonic frequencies?

Ok, so now you are playing a 440Hz (A4) saw wave, it contains 440Hz, 880Hz, 1320Hz etc. Anything generated using the signal generator can contain only these frequencies, the only difference is the levels and phases of each of them. What if you want to make the signal dirty by adding say 500Hz? Well, that is not that simple! Here we are getting into audio synthesizer stuff, so let us just give you a few hints.

The traditional way is to use modulation. One particular method is called frequency modulation (FM). Instead of generating a 440Hz saw wave with your generator, you change the pitch, up and down. You are modulating the frequency, that's why FM. It is basically a vibrato, but as you increase the speed of the vibrato, it gets so quick that you stop noticing the pitch changes (that's very simplified but it serves the purpose) and instead it starts producing a very complex spectrum. Will the 500Hz be there? Well, if setup correctly, yes, but there will also be lots of other non-harmonic frequencies.

Another way is possible without any other tools. Let's say you do not want 440Hz, but 660Hz. Then you may generate 220Hz instead of 440Hz (which is one octave below it) and voila, 660Hz is the 3rd harmonic (3 x 220 is 660)! But you need to shift the saw wave one octave above. Fortunately it is not that hard here - go to the normal mode, select saw tooth, click advanced, and use the harmonics panel to remove the fundamental and leave just the 2nd harmonic, then convert it to harmonic mode. Well, it's not that hard, but it's not exactly simple either...

The only way is, of course, additive synthesis. In that case you do not use one oscillator, but many of them. It lets you generate just about anything. But there is a catch, actually many of them. First, you need to say "ok I want this frequency and that frequency...", the setup is actually infinitely hard as there may be an infinite number of frequencies :). And the second is, of course, CPU requirements.

So is there some ultimate solution? Nope, sorry. The good thing is, you will not probably need it, because while what you see is not always what you get, also what you want is often not what you really want to hear :).



### Random button

Random button generates random settings using the existing presets.



### Presets button

Presets button displays a window where you can load and manage available presets.

Hold **Ctrl** when clicking to load a random preset instead.

### ◀ **Left arrow button**

Left arrow button loads the previous preset.

### ▶ **Right arrow button**

Right arrow button loads the next preset.

### 🎲 **Randomize button**

Randomize button loads a random preset.

### 📄 **button**

This button copies the settings onto the system clipboard.

### 📄 **button**

This button loads the settings from the system clipboard.

### 🎲 **Randomize button**

Randomize button generates random settings.

NORMAL

### **Normal button**

Normal button switches the generator into the normal mode, which lets you edit the shape of the oscillator. This is especially advantageous for low-frequency oscillators, where the shape matters even though it doesn't have any physical meaning.

### ⬇️ **Convert button**

Convert button converts the current shape into harmonic-based representation. Please note that since the number of harmonics is limited, the result will not perfectly resemble the original shape.

HARMONICS

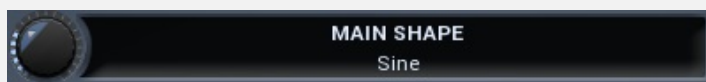
### **Harmonics button**

Harmonics button switches the generator into the harmonics mode, which lets you edit the levels and phases of individual harmonics. This is especially advantageous for high-frequency oscillators, hence sound generators.

## Signal generator in Normal mode

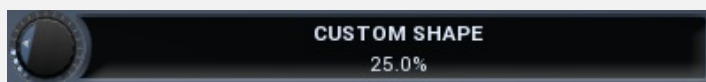


Signal generator in Normal mode works by generating the oscillator shape using a combination of several curves - a predefined set of standard curves, custom shape, step sequencer and custom sample. It also post-processes the shape using several filters including smoothing to custom transformations. This is especially useful when using the oscillator as an LFO (low-frequency-oscillator), where the harmonic contents does not really matter, but the shape does.



### Main shape

Main shape controls the main shape used by the signal generator. There are several predefined shapes, such as sine, triangle or rectangle, which you can choose from or even interpolate between using this control.



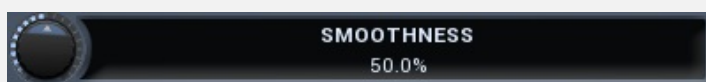
### Custom shape

Custom shape controls the amount of the custom shape that is blended into the main shape.



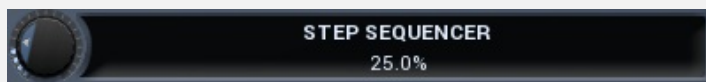
### Edit button

Edit button shows the custom shape editor.



### Smoothness

Smoothness controls the amount of smoothing. Many shapes, especially those produced by the step sequencer, have rough jagged edges, which may be advantageous, but when used to modulate certain parameters, the output may be clicking or causing other artifacts. Smoothness helps it by smoothing the whole signal shape out and removing these rough edges.



### Step sequencer

Step sequencer controls the amount of the step sequencer shape that is blended into the main shape (which has already been blended with the custom shape).



### Edit button

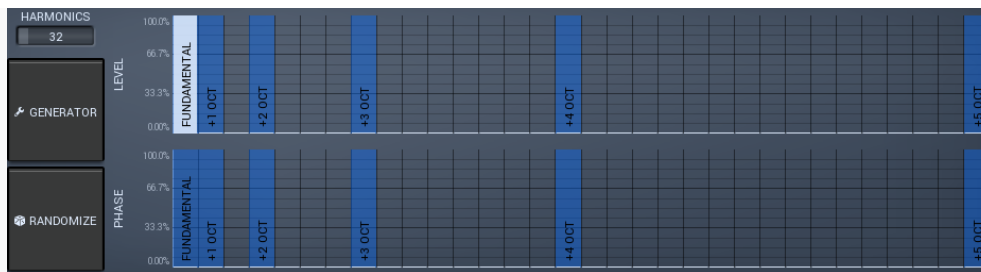
Edit button shows the step sequencer editor.



### Advanced button

Advanced button displays an additional window with more advanced settings for post-processing the signal shape, such as harmonics or custom transformations.

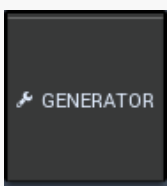
## Signal generator in Harmonics mode



Signal generator in Harmonics mode works by generating the oscillator shape using individual harmonics. Essentially a harmonic is a sine wave. The first harmonic, known as the fundamental, fits once in the oscillator time period, hence it is the same as selecting sine wave in the **Normal mode**. The second harmonic fits twice, the third three times etc. In theory, any shape you create in normal mode can be converted into harmonics. However, this approach to signal generation needs an enormous number of harmonics, which is both inefficient to calculate and mostly hard to edit. Therefore, the harmonic mode can process up to 256 harmonics, which is enough for very complex spectrums, however it is still not enough to generate an accurate square wave for example. If your goal is to create basic shapes, it is better to use the normal mode.

It is nearly impossible to say how a particular curve will sound when used as a high-frequency oscillator in a synthesizer, just by looking at its shape. Harmonics mode, on the other hand, is directly related to human hearing and makes this process very simple. In general, the more harmonics you add, the richer the sound will be. The higher the harmonic, the higher the tone. Usually, one leaves the first harmonic enabled too, as this is the fundamental tone, however you may experiment with more dissonant sounds without it.

Editing harmonics can be time consuming unless you hear what you want, so a signal generator is also available. This great tool lets you generate a random spectrum by a single click. You can also open the **Generator<reference> settings and edit its parameters, which basically control the audio properties in a more natural way - using parameters such as complexity, harmonicity etc.**



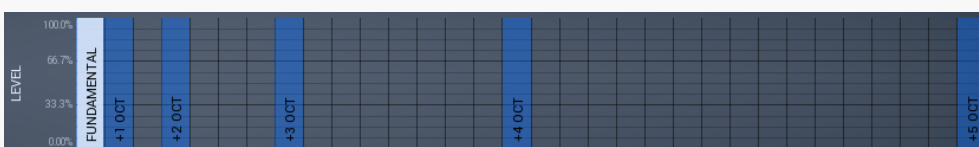
### button

This button shows a powerful harmonics generator, which can create unlimited number of various timbres and even analyze a sample and extract harmonics from it.



### Randomize button

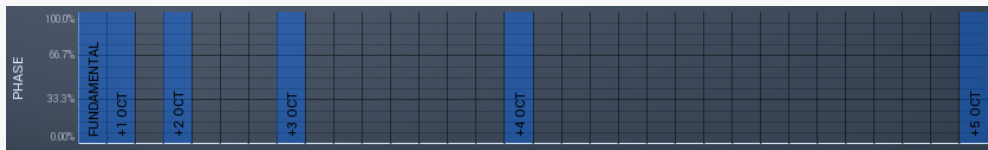
Randomize button selects random parameters for the harmonics generator, so you can use it to get a random sound character instantly. Hold **Ctrl** to slightly modify existing generator settings instead of completely changing them.



### Magnitudes

## graph

Magnitudes graph contains the levels of the individual harmonics. The highlighted bars are octaves, thus the 1st, 2nd, 4th, 8th harmonic etc.



## Phases graph

Phases graph contains the phases of the individual harmonics. The highlighted bars are octaves, thus the 1st, 2nd, 4th, 8th harmonic etc.

## Osc 2/3 panel



Osc 2/3 panel contains settings for the secondary oscillator. There are 2 secondary oscillators, neither of them has the unison feature, however these 2 have combination mode instead. By default the oscillator is simply mixed with the current signal, however the mode lets you perform somewhat more complicated transformation, such as frequency modulation or convolution.

### **Random button**

Random button generates random settings using the existing presets.

### **Presets button**

Presets button displays a window where you can load and manage available presets. Hold **Ctrl** when clicking to load a random preset instead.

### **Left arrow button**

Left arrow button loads the previous preset.

### **Right arrow button**

Right arrow button loads the next preset.

## **Randomize button**

Randomize button loads a random preset.

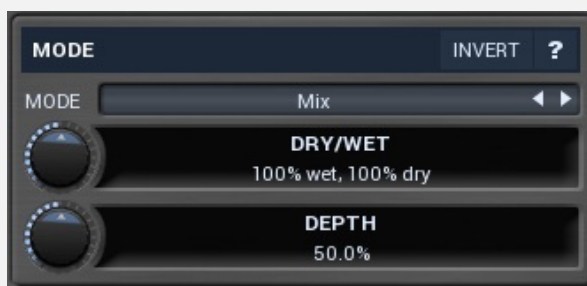
## **button**

This button copies the settings onto the system clipboard.

## **button**

This button loads the settings from the system clipboard.

# Mode panel



Mode panel contains parameters controlling how the generator is used.

## **Invert**

Invert switch inverts the oscillator output. This may be handy if multiple oscillators are cancelling each other.

## **Mode**

Mode controls the way the generator is used.

**Mix** mode simply mixes the 2 signals, main input and modulator. Here, as below, the main input is the audio from the previous section and the modulator is the audio from this oscillator.

**Frequency modulation** performs a frequency modulation of main input signal by the modulator. This is essentially an extremely fast vibrato, which produces a very rich spectrum of harmonic and non-harmonic frequencies. Depth parameter controls the range of the vibrato, so the higher the depth is, the stronger the modulation. In effect, the depth parameter has the same effect as level of the modulator signal.

**Frequency modulation (abs)** is similar to the frequency modulation, but it provides an even richer spectrum and has a special feature - if the modulation signal is 0 (silence), the output is not changed at all. This is not true for the traditional frequency modulation, which takes effect even if the input is zero.

This may be useful when the modulation signal is temporary, mostly silent, and you don't want the input altered at all in the silent parts. A typical example is click simulation - most keyboard instruments "click" when a key is pressed. The click is very rich in spectrum and temporary obviously, which makes the frequency modulation (abs) an ideal candidate.

**Ring modulation** multiplies the 2 signals together, which results in so-called ring modulation effect, which usually provides a very rich output spectrum, however in



most cases it is not harmonic any more, often sounding metallic. It is often a good idea to make one of the generators produce very low frequencies when using ring modulation. Depth parameter serves as a dry/wet ratio between the main input signal and output of the ring modulation.

The basic principle is that when you ring-modulate 2 sine waves with frequencies  $F1$  and  $F2$ , the output will contain frequencies  $F1+F2$  and  $F1-F2$ . It's therefore useful to have one of the frequencies (usually  $F2$ , called the modulator) low enough, even say 20Hz. That way the resulting frequencies will lie close to the original one and may enrich the signal.

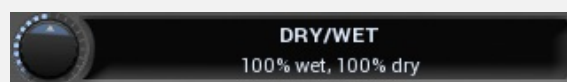
An interesting thing happens when the 2 frequencies are an octave (or multiple octaves) apart. For example, if  $F1$  is 440Hz (A4) and  $F2$  is 220Hz (A3), then the resulting frequencies are  $440-220=220$ Hz (again A3) and  $440+220=660$  (E5), which is nothing else than the 3rd harmonic. In this case the output of the ring modulation is still harmonic and provides an interesting way to enrich the signal. In practice you'll like to use more complex waveforms than sine waves, which leads to quite more complicated results.

**Frequency shifting** is similar to ring modulation, but it performs some complicated filtering to remove the lower output frequency,  $F1-F2$ . In practice this is not completely possible, but it significantly attenuates the lower frequencies making the output cleaner. The name, frequency shifting is descended from the sole purpose - shifting all frequencies upwards or downwards. This is a quite different output from pitch shifting as it doesn't keep harmonic relationships between frequencies within the signal.

**Maximization** takes the maximum of both signals. This usually causes some form of dirtiness or distortion and in a way contains at least all frequencies from both signals. Hence this is yet another way to enrich them. Depth parameter serves as a dry/wet ratio between the main input signal and output of the maximization.

**Convolution** is inspired by traditional convolution and produces similar results. Whereas other modes make the signal richer by combining the spectral components of both signals in some way, convolution does the opposite - it enhances the spectral components present in both and attenuates those which are present in just one of the input signals. In other words, it filters one signal by another. Usually you will want to process a rich signal with another rich signal with this mode. Depth parameter controls the convolution length, which basically defines how deep the modulation will be. Note that the higher the depth, the more CPU power is needed.

**Sync** simulates hard synchronization - it resets the modulator every time the main input starts a new period. In this case the main input and modulator are swapped. That way the modulator controls the output spectrum, while main input controls the pitch. This kind of modulation provides reasonable results only for simple waveforms, the best start would be any simple wave (square, triangle, sine...) as main input and saw modulator. Then use modulator pitch and shape to change the sound character.

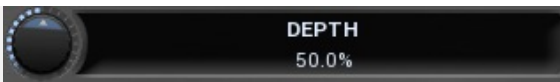


## Dry/wet

Dry/wet defines the mix ratio for this oscillator.

Range: 0% wet, 100% dry to 100% wet, 0% dry, default 100% wet, 100% dry





## Depth

Depth controls the amount of modulation caused by this generator. It is used in some modes only.

Range: 0.00% to 100.0%, default 50.0%

## Pitch panel



Pitch panel lets you shift the generator pitch.

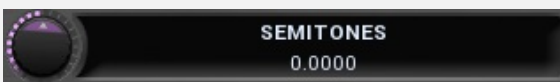
### CONSTANT FREQUENCY **Constant frequency**

Constant frequency makes the oscillator ignore the note pitch and behave as if an A4 (440Hz) note is received each time.

### OCTAVES **Octaves**

Octaves defines the generator pitch change (in octaves).

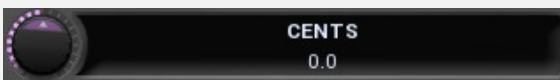
Range: -8 to +8, default 0



### **Semitones**

Semitones defines the generator pitch change (in semitones).

Range: -24.0000 to +24.0000, default 0.0000

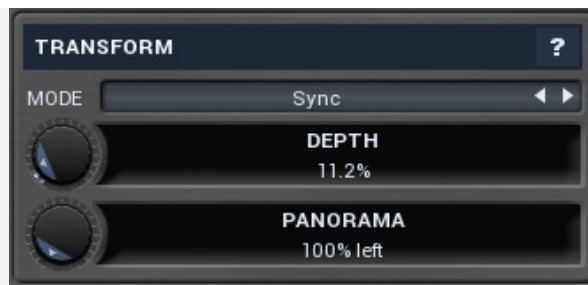


### **Cents**

Cents defines the generator pitch change (in cents of a semitone). The actual pitch change is the sum of these 3 control values.

Range: -100.0 to +100.0, default 0.0

## Transform panel



Transform panel contains some additional oscillator transformation features.

## MODE Sync ◀ ▶ **Mode**

Mode controls the type of transformation used on the oscillator waveform. The oscillators themselves are pre-processed in a very complex manner to provide maximum audio quality. However that means that every time you change something in the oscillator editor, lots of calculations need to be performed, which may take quite some CPU. The transformations however are performed without pre-processing, so they may provide lower audio quality, but they are much less CPU intensive and often provide actually higher quality when modulated, because these are always sample accurate.

It is therefore recommended to use the oscillator editor for static sounds and then use transformations for modulation. Another reason to use the transformation is the actual harmonic properties of the generated sound - any oscillator (algorithmic or wavetable) produces only a fundamental frequency and its harmonics, the actual wave shape only controls the levels and phases of these frequencies. This means that if the fundamental pitch is say 100Hz, then the generated sound contains only 100Hz, 200Hz, 300Hz etc. It can never produce 150Hz for example. And that's when the transformations comes in handy again. When you transform the waveform in any way the result is just another waveform, so it still cannot produce other frequencies. However if you change the transformation depth in real-time, say using modulators, the output could produce just about any frequencies, all depending on the actual transformation and the way it is modulated.

**Disabled** disables any transformation.

**PWM** implements several pulse-width modulation algorithms, all affecting the width of the pulse or bending its shape in time. PWM tends to produce frequencies around each harmonic when modulated. The quicker the PWM depth is changed, the further these frequencies are from the main harmonics and the sound gets richer until a point where it is completely disharmonic.

**Sync** essentially duplicates the periods of the waveform (not necessarily a multiple). The higher the depth, the higher the dominant harmonic, so moving the depth in a way controls the pitch without losing the sense of the actual fundamental pitch. Sync waveform has abrupt edges in most cases. These generate lots of higher harmonic content, sometimes even reaching zipper noise. To diminish this and to get a cleaner sound, use the windowed versions, which smooth out the edges.

**ASync** transformations are similar to Sync, however the waveforms are bent, so the typical sync screaming is less resonant and more inharmonic.

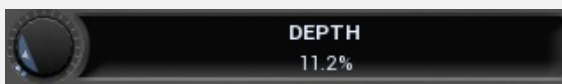
**Bending and mirroring** alter the waveform quite smoothly, so these produce less rich tones and the effects may remind you of phasing. Discontinuities may occur when modulated quickly, so you can exploit that to get a richer sound if needed.

**Reversing** merges the waveform with a reversed replica of itself. This may often cause abrupt edges producing a sort of a modulated square wave. To get the most

out of it, the original waveform should not be (anti)symmetric - when you look at a sine wave for example, the right half is just an inverted left half, so reversing it isn't such a miraculous action. This is true for most basic waves including saw and rectangle for example. Therefore reversing is most interesting when used with some more complex wave-shapes produced by harmonic mode or the step sequencer etc. **Invert, Maximize, Zero under, Power and Quantize** are manipulating the amplitude only and due to the discontinuities these produce fairly rich sounds, lo-fi in a way. Please note the processor may produce different results for different quality modes, because it can generate specific waveforms for different notes to improve quality, but since the amplitude will be different, the results of the transformation will be different as well.

**AM** provides amplitude modulation with itself or a reversed version of itself. As such it provides minimum non-harmonic frequencies.

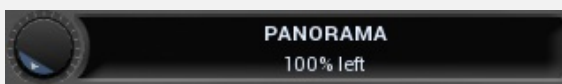
**Recursive** transformations are the most complex and basically transform the waveform by projecting it onto itself. The output generally depends on the input waveform signal and you might say that the more complex the initial waveform is, the more complex the output waveform will be. Recursive transformations may produce extremely full spectra and may mix the properties and richness of other transformations combined. As such, simple waveforms such as sine and similar usually produce the best results. Modulating a harmonically complex waveform can easily end up with white noise.



## Depth

Depth controls the amount/type of selected transformation and shall be used for modulation. Its meaning depends on the selected transformation.

Range: 0.00% to 100.0%, default 50.0%



## Panorama

Panorama defines the oscillator panorama transformation.

Range: 100% left to 100% right, default center

# Signal editor



Signal editor controls the oscillator shape. You can use either oscillator shapes or harmonics; the latter are preferred in this case as they directly control the harmonic structure of the generated sounds. Signal-generator is an incredibly versatile generator of low & high frequency signals. It offers 2 distinct modes - Normal and

Harmonics.

**Normal mode** is appropriate for low-frequency oscillators, where the graphical shape is relevant and is used to drive some form of modulation. For example, a tremolo uses this modulation to change the actual signal level in time. Frequencies for such oscillators usually do not exceed 20Hz as this is a sort of limit above which the frequencies become audible.

**Harmonics** mode is designed for high-frequency oscillators, where the actual shape is not as important as the harmonic content of the resulting signal, hence it is especially useful for actual audio signals. Please note that since a shape can contain more harmonics than those available from the harmonic generator, the results may not be exactly the same. As an example, a rectangular wave in normal mode may sound fuller than when converted to the harmonic mode.

Use the arrow-down button to switch from normal mode to harmonics mode or click the **Normal** and **Harmonics** buttons

## Normal mode

The generator first uses a set of predefined signal shapes (sine, triangle, rectangle...), which you can select directly by right-clicking on the editor and choosing the requested shape from the menu. This menu also provides a link to the modulator shapes preset manager, normalization and randomization. You can also use the **Main shape** parameter, which generates a combination of adjacent signals to provide a nearly inexhaustible number of basic shapes.

The engine then combines the predefined shape with a **Custom shape**, which may be anything you can draw using the advanced envelope engine, depending on the level set by the **Custom shape** control. Use the **Edit** button to edit the custom shape.

You can also combine those results with a fully featured step sequencer, with variable number of steps and several shapes for each of them, depending on the level set by the **Step sequencer** control. Use the lower **Edit** button to edit the step sequence.

Those results may be mixed with a custom sample, which is available from the advanced settings, accessed by clicking the **Advanced** button.

**Smoothness** softens any abrupt edges, generated by the step sequencer for example.

Finally there are **Advanced** features providing more complex transformations, adding harmonics etc. or you can click the **Randomize** button in the top-left corner to generate a random, but reasonable, modulator shape.

## Harmonics mode

Harmonics mode represents the signal as a series of harmonics (that is, multiples of the base frequency). For example, when your oscillator has a frequency of 2Hz (set in the panel), then the harmonics are 2Hz, 4Hz, 6Hz, 8Hz etc. In theory, any signal can

be created by mixing a potentially infinite number of these harmonics.

The harmonics mode lets you control the levels and phases of each harmonic. The top graph controls the levels of individual harmonics, while the bottom one controls their phases. Use the left-mouse button to change the values in each graph, the right-mouse button sets the default for the harmonics - 0% level and 0% phase. In both graphs the harmonics of power 2 (that is octaves) are highlighted. Other harmonics may actually sound disharmonic, despite their names.

*For example, if you reset all harmonics to the defaults and increase only the first one, you will get a simple sine wave. By adding further harmonics you make the output signal more complex.*

**Harmonics** controls the number of generated harmonics. The higher the number is, the richer the output signal is (unless the levels are 0% of course). This is useful to make the sound cleaner. For example, if you transform a saw-tooth wave to harmonics, it would not sound like a typical saw-tooth wave anymore, but more like a low-passed version of one. The more harmonics you use, the closer you get to the original saw-tooth wave.

**Generator** is a powerful tool for generating the harmonics, which are otherwise rather clumsy to edit. The generator provides several parameters based upon which it creates the entire series of harmonic levels and phases. These parameters are usually easier to understand than the harmonics themselves. Part of the generator is the randomizer available via the **Random seed** button, which smartly generates random settings for the generator. This makes the process of getting new sounds as simple as possible.

## Signal generation fundamentals

The signal generator produces a periodic signal with specified wave shape. This means that the signal is repeating over and over again. As a result it can only contain multiples of the fundamental frequency. For example, if the generator is producing 100Hz signal, then it can contain 100Hz (fundamental or 1st harmonic), 200Hz (2nd harmonic), 300Hz (3rd harmonic), 400Hz (4th harmonic) etc. However, it can never produce 110Hz. You can then control the level of each harmonic and their relative phases. It does not matter whether you use the normal mode using oscillator shapes, or harmonics mode where you can control the harmonics directly. If both modes result in the same wave shape (such as sine wave vs. 1st harmonic only), then the result is exactly the same.

Sine wave is the simplest of all as it contains the fundamental frequency only. The "sharper" the signal shape is, the more harmonics it contains. The biggest source of higher harmonics is a "discontinuity", which you can see in both rectangle and saw waves. In theory, these signals have an infinite number of harmonics. However since our hearing is highly limited to less than 20kHz, the number of harmonics which are relevant is actually pretty small. If you generate a 50Hz signal, which is very low, and assuming that you have extremely good ears and you actually hear 20kHz, then the number of harmonics audible for you is  $20000 / 50 = 400$ .

## What happens above 20kHz?

Consider the example above again, what happens with harmonics above 400? These either stay there and simply are not audible, disappear if anti-aliasing is used, or get aliased back under 20kHz in which case you get the typical digital dirt.

When you convert a rectangle wave to harmonics mode, only the first 256 harmonics are used, so it basically works like an infinitely steep low-pass filter. What is the limit then?  $50 \text{ Hz} * 256 = 12.8\text{kHz}$ . The harmonic mode will not produce anything above this limit if you are generating a 50Hz signal. Most people do not hear anything above 15kHz, so this is usually enough, but if not, you may need to use the normal mode where you get the "infinite" number of harmonics.

## What you see is not always what you get!

Say you want a rectangle wave and play a 440Hz tone(A4). You would expect the output signal to be a really quick rectangle wave, right? Wrong! If you would do that, and actually most synthesizers on the market do that, you would get the infinite number of harmonics. And, since you are working in say 48kHz sampling rate, the maximum frequency that can actually exist in your signal is 24kHz. So everything above it would get aliased below 24kHz, and there would be a lot of aliased dirt.

The "good" synthesizers perform a so-called anti-aliasing. There are several methods, most of them require quite a lot of CPU or have other limitations. The goal is to remove all frequencies above the 24kHz in our case or in reality, it is more about removing all aliased frequencies above 20kHz - this means, that we do not care about frequencies above 20kHz, because we do not hear them anyway. But we will keep it simple. Let's say we remove everything above 20kHz. You already know that the rectangle wave can be created using an infinite number of harmonics or sine waves. We removed everything above the 45th harmonic ( $20000 / 440$ ) so our rectangle wave is trying to be formed using just 45 harmonics, so it will not really look like a rectangle wave.

After some additional filtering (like DC removal), the rectangle wave may look completely different than a true rectangle wave, yet it would sound the same! Does it matter? Not really. You simply edit the shape as a rectangle wave and let the synthesizer do the ugly stuff for you. But do not check the output, because it may be very different than what you would expect ;).

## How can I generate non-harmonic frequencies?

Ok, so now you are playing a 440Hz (A4) saw wave, it contains 440Hz, 880Hz, 1320Hz etc. Anything generated using the signal generator can contain only these frequencies, the only difference is the levels and phases of each of them. What if you want to make the signal dirty by adding say 500Hz? Well, that is not that simple! Here we are getting into audio synthesizer stuff, so let us just give you a few hints.

The traditional way is to use modulation. One particular method is called frequency modulation (FM). Instead of generating a 440Hz saw wave with your generator, you change the pitch, up and down. You are modulating the frequency, that's why FM. It is basically a vibrato, but as you increase the speed of the vibrato, it gets so quick that

you stop noticing the pitch changes (that's very simplified but it serves the purpose) and instead it starts producing a very complex spectrum. Will the 500Hz be there? Well, if setup correctly, yes, but there will also be lots of other non-harmonic frequencies.

Another way is possible without any other tools. Let's say you do not want 440Hz, but 660Hz. Then you may generate 220Hz instead of 440Hz (which is one octave below it) and voila, 660Hz is the 3rd harmonic ( $3 \times 220$  is 660)! But you need to shift the saw wave one octave above. Fortunately it is not that hard here - go to the normal mode, select saw tooth, click advanced, and use the harmonics panel to remove the fundamental and leave just the 2nd harmonic, then convert it to harmonic mode. Well, it's not that hard, but it's not exactly simple either...

The only way is, of course, additive synthesis. In that case you do not use one oscillator, but many of them. It lets you generate just about anything. But there is a catch, actually many of them. First, you need to say "ok I want this frequency and that frequency...", the setup is actually infinitely hard as there may be an infinite number of frequencies :). And the second is, of course, CPU requirements.

So is there some ultimate solution? Nope, sorry. The good thing is, you will not probably need it, because while what you see is not always what you get, also what you want is often not what you really want to hear :).

### **Random button**

Random button generates random settings using the existing presets.

### **Presets button**

Presets button displays a window where you can load and manage available presets. Hold **Ctrl** when clicking to load a random preset instead.

### **Left arrow button**

Left arrow button loads the previous preset.

### **Right arrow button**

Right arrow button loads the next preset.

### **Randomize button**

Randomize button loads a random preset.

### **button**

This button copies the settings onto the system clipboard.

### **button**

This button loads the settings from the system clipboard.

### **Randomize button**

Randomize button generates random settings.



## Normal button

Normal button switches the generator into the normal mode, which lets you edit the shape of the oscillator. This is especially advantageous for low-frequency oscillators, where the shape matters even though it doesn't have any physical meaning.



## Convert button

Convert button converts the current shape into harmonic-based representation. Please note that since the number of harmonics is limited, the result will not perfectly resemble the original shape.

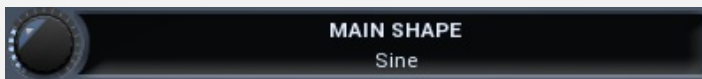
## Harmonics button

Harmonics button switches the generator into the harmonics mode, which lets you edit the levels and phases of individual harmonics. This is especially advantageous for high-frequency oscillators, hence sound generators.

## Signal generator in Normal mode

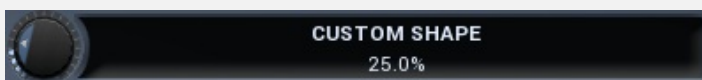


Signal generator in Normal mode works by generating the oscillator shape using a combination of several curves - a predefined set of standard curves, custom shape, step sequencer and custom sample. It also post-processes the shape using several filters including smoothing to custom transformations. This is especially useful when using the oscillator as an LFO (low-frequency-oscillator), where the harmonic contents does not really matter, but the shape does.



### Main shape

Main shape controls the main shape used by the signal generator. There are several predefined shapes, such as sine, triangle or rectangle, which you can choose from or even interpolate between using this control.



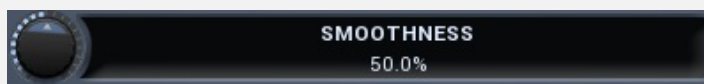
### Custom shape

Custom shape controls the amount of the custom shape that is blended into the main shape.



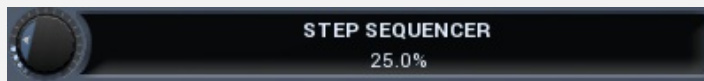
### Edit button

Edit button shows the custom shape editor.



## Smoothness

Smoothness controls the amount of smoothing. Many shapes, especially those produced by the step sequencer, have rough jagged edges, which may be advantageous, but when used to modulate certain parameters, the output may be clicking or causing other artifacts. Smoothness helps it by smoothing the whole signal shape out and removing these rough edges.



## Step sequencer

Step sequencer controls the amount of the step sequencer shape that is blended into the main shape (which has already been blended with the custom shape).



## Edit button

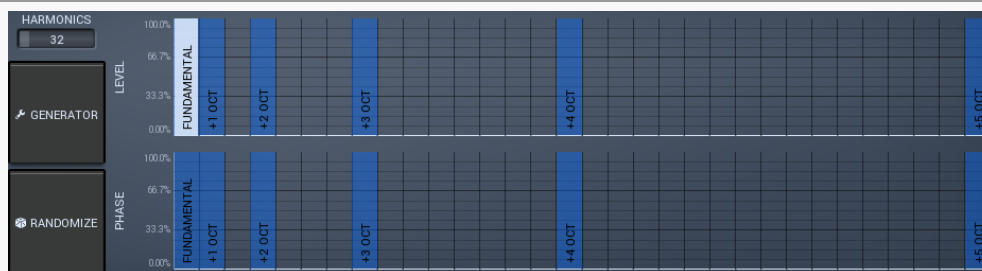
Edit button shows the step sequencer editor.



## Advanced button

Advanced button displays an additional window with more advanced settings for post-processing the signal shape, such as harmonics or custom transformations.

# Signal generator in Harmonics mode

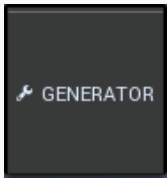


Signal generator in Harmonics mode works by generating the oscillator shape using individual harmonics. Essentially a harmonic is a sine wave. The first harmonic, known as the fundamental, fits once in the oscillator time period, hence it is the same as selecting sine wave in the **Normal mode**. The second harmonic fits twice, the third three times etc. In theory, any shape you create in normal mode can be converted into harmonics. However, this approach to signal generation needs an enormous number of harmonics, which is both inefficient to calculate and mostly hard to edit. Therefore, the harmonic mode can process up to 256 harmonics, which is enough for very complex spectrums, however it is still not enough to generate an accurate square wave for example. If your goal is to create basic shapes, it is better to use the normal mode.

It is nearly impossible to say how a particular curve will sound when used as a high-frequency oscillator in a synthesizer, just by looking at its shape. Harmonics mode, on the other hand, is directly related to human hearing and makes this process very simple. In general, the more harmonics you add, the richer the sound will be. The higher the

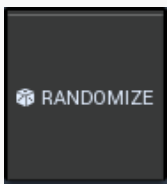
harmonic, the higher the tone. Usually, one leaves the first harmonic enabled too, as this is the fundamental tone, however you may experiment with more dissonant sounds without it.

Editing harmonics can be time consuming unless you hear what you want, so a signal generator is also available. This great tool lets you generate a random spectrum by a single click. You can also open the **Generator<\reference> settings and edit its parameters, which basically control the audio properties in a more natural way - using parameters such as complexity, harmonicity etc.**



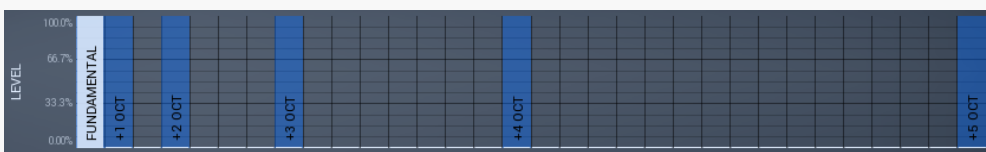
## button

This button shows a powerful harmonics generator, which can create unlimited number of various timbres and even analyze a sample and extract harmonics from it.



## Randomize button

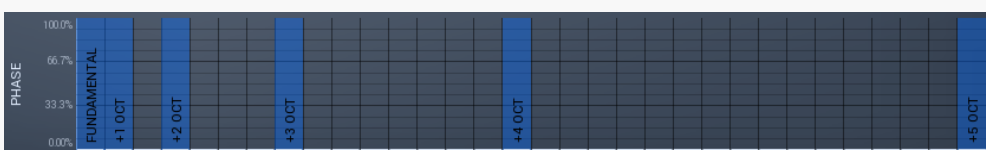
Randomize button selects random parameters for the harmonics generator, so you can use it to get a random sound character instantly. Hold **Ctrl** to slightly modify existing generator settings instead of completely changing them.



## Magnitudes

### graph

Magnitudes graph contains the levels of the individual harmonics. The highlighted bars are octaves, thus the 1st, 2nd, 4th, 8th harmonic etc.



## Phases graph

Phases graph contains the phases of the individual harmonics. The highlighted bars are octaves, thus the 1st, 2nd, 4th, 8th harmonic etc.

# Noise panel



Noise panel contains parameters of the noise generator, which can be mixed with the main signal. The noise generator can also have its own ADSR envelope.

### **Random button**

Random button generates random settings using the existing presets.

### **Presets button**

Presets button displays a window where you can load and manage available presets. Hold **Ctrl** when clicking to load a random preset instead.

### **Left arrow button**

Left arrow button loads the previous preset.

### **Right arrow button**

Right arrow button loads the next preset.

### **Randomize button**

Randomize button loads a random preset.

### **button**

This button copies the settings onto the system clipboard.

### **button**

This button loads the settings from the system clipboard.

### **Mode**

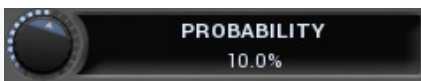
Mode defines the type of output randomized signal. **White noise** generates standard white noise, which has equal energy in each frequency. **Pink noise** provides an accurate approximation of pink noise, where the energy is decreasing by 3dB per octave. **Clicks & pops** define generated random peaks with specified probability. **Stairs** is similar to clicks & pops, but after each peak the generated value is following instead of generating zeros. Hence, the resulting waveform looks like random stairs.



### **Dry/wet**

Dry/wet defines the mix ratio for the noise generator.

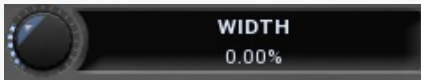
Range: 0% wet, 100% dry to 100% wet, 0% dry, default 100% wet, 100% dry



## Probability

Probability defines a probability coefficient used in some of the modes somehow. For example for clicks & pops mode it affects how often the clicks happen.

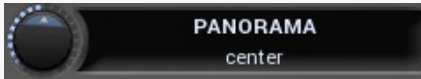
Range: 0.00% to 100.0%, default 10.0%



## Width

Width controls the noise stereo width.

Range: Mono to 200.0%, default 0.00%



## Panorama

Panorama defines the noise generator panorama transformation.

Range: 100% left to 100% right, default center

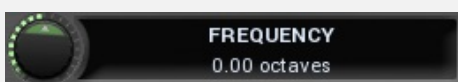
# High-pass



High-pass contains parameters of the noise generator high-pass filter.

## CONSTANT FREQUENCY Constant frequency

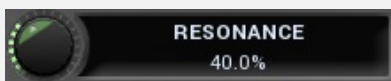
Constant frequency makes the filter ignore the note pitch and behave as if an A4 (440Hz) note is received each time.



## Frequency

Frequency controls the high-pass filter frequency.

Range: -8.00 octaves to +8.00 octaves, default 0.00 octaves



## Resonance

Resonance controls the high-pass filter resonance.

Range: 0.00% to 100.0%, default 40.0%



## Slope

Slope controls the high-pass filter slope.

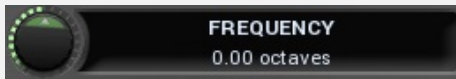
# Low-pass



Low-pass contains parameters of the noise generator low-pass filter.

### CONSTANT FREQUENCY **Constant frequency**

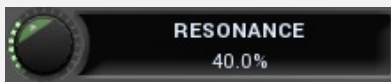
Constant frequency makes the filter ignore note pitch and behave like A4 (440Hz) has always been received.



### **Frequency**

Frequency controls the low-pass filter frequency.

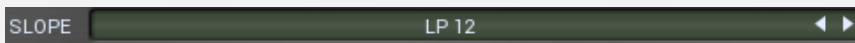
Range: -8.00 octaves to +8.00 octaves, default 0.00 octaves



### **Resonance**

Resonance controls the low-pass filter resonance.

Range: 0.00% to 100.0%, default 40.0%



### **Slope**

Slope controls the low-pass filter slope.

## ADSR graph



ADSR graph controls the ADSR envelope for the noise generator. Our ADSR envelopes are much more sophisticated than classic attack-release-sustain-decay envelopes.

Besides these common parameters they also let you control the curvature of each stage. Additionally, there are hold and delay sections, global smoothing and tremolo. You can even use the custom shape mode to define your own attack/release curves.

### **RANDOM button**

Random button generates random settings using the existing presets.

### **Presets button**

Presets button displays a window where you can load and manage available presets. Hold **Ctrl** when clicking to load a random preset instead.

## ◀ Left arrow button

Left arrow button loads the previous preset.

## ▶ Right arrow button

Right arrow button loads the next preset.

## 📄 button

This button copies the settings onto the system clipboard.

## 📄 button

This button loads the settings from the system clipboard.

🔗 Custom shape

## Custom shape button

Custom shape button enables custom shape mode, which lets you draw your own attack and release stages using the envelope system. Both stages are then automatically connected to form the resulting envelope.

⏏ Immediate release

## Immediate release button

Immediate release button activates the immediate release mode in which case the note-off causes an immediate switch to the release stage. If this is disabled, the release stage does not occur until the whole attack/decay stage finishes.

🔄 Sync

## Sync button

Sync button controls the ADSR tempo sync feature. By default this is disabled and means that all times are followed exactly, meaning that if **Attack** is say 100ms, then it will be 100ms indeed. Tempo sync lets the plugin adjust the times to ensure it will be always in sync with the host tempo. In this case 100ms may become say 125ms if the tempo is 120bpm, because 125ms is the length of a 16th note. This makes it extremely simple to convert any envelope to a tempo-synced one. The plugin always chooses the nearest longer note, in other words it always round up.

**Straight** and **Triplets** modes automatically find 'nice' values.

*For example, if a 16th note takes 100ms, the attack time is 550ms, and the sync mode is straight, then the plugin checks for 100ms, find out that it is too low, so it checks 8th note, being 200ms, still too low, then continues with quarter note, which takes 400ms, and still not enough, finally 800ms corresponding to a half note is the one, so the resulting time will be 800ms. Triplet cases are more complex, but the principle is the same.*

**1/16**, **1/8** and **1/4** modes choose the nearest higher multiply of the base note length. For example, if a 16th note takes 100ms, the attack time is 550ms, and the sync mode is 1/16, the resulting time will be 600ms.

🌀 TREMOLO

## Tremolo button

Tremolo button displays additional tremolo settings, containing tremolo behaviour and shape.

PREVIEW

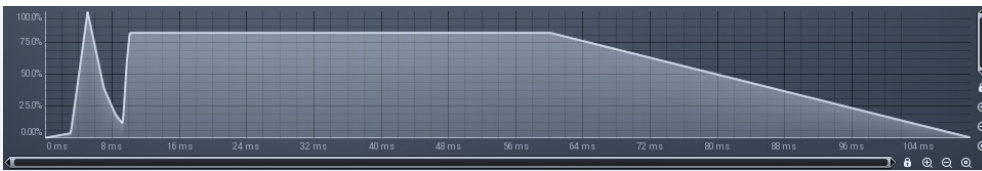
ATTACK

RELEASE

## Selector

Selector lets you select the current part of the custom envelope. **Preview** displays the resulting envelope. **Attack** lets you edit the attack part of the envelope and **Release** lets you edit the release part.





## Envelope

### graph

Envelope graph contains the preview/attack/release stages. Use this to edit your custom ADSR envelope.

# Filter panel



Filter panel contains settings for the particular filter.

### **Random button**

Random button generates random settings using the existing presets.

### **Presets button**

Presets button displays a window where you can load and manage available presets. Hold **Ctrl** when clicking to load a random preset instead.

### **Left arrow button**

Left arrow button loads the previous preset.

### **Right arrow button**

Right arrow button loads the next preset.

### **Randomize button**

Randomize button loads a random preset.

## **button**

This button copies the settings onto the system clipboard.

## **button**

This button loads the settings from the system clipboard.

## **Clip**

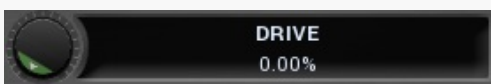
Clip enables an optional clipper placed after the filter's output gain. It can be used to remove potential peaks that the filter may cause especially with wild settings. As a hard clipper it can also be used as another distortion stage, however be cautious with it.

## **Constant frequency**

Constant frequency makes the filter ignore the note pitch and behave as if an A4 (440Hz) note is received each time. You can still use the velocity/note tracking when this is enabled.

## **Parallel**

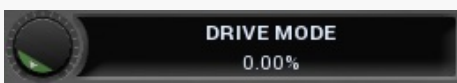
Parallel enables parallel processing for the filters. Normally each filter processes the output of the previous filter and its output becomes the new signal. If parallel processing is enabled, all filters process the original generator signal instead and their outputs are mixed together.



## **Drive**

Drive controls the input distortion of the filter. This creates higher harmonics, which are then processed together with the original signal through the filter. Usually the drier the input signal is, the more drive may be used to make the signal richer before any filtering occurs. When applied to an already-rich signal, the results may simply be too dirty. However when applied to a rich yet harmonic signal (such as a saw-tooth wave), only existing harmonics will be added, so the effect won't be creating additional harmonics but rather changing their levels resulting in a different spectrum. It is highly advised to use the plugin's upsampling feature in order to minimize disharmonic components created by aliasing.

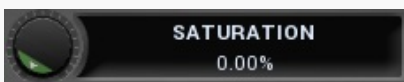
Range: 0.00% to 100.0%, default 0.00%



## **Drive mode**

Drive mode controls the input distortion character. Essentially this controls the levels of different harmonics.

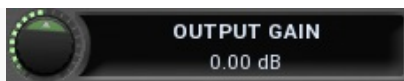
Range: 0.00% to 100.0%, default 0.00%



## **Saturation**

Saturation controls the output saturation that is performed before the output gain. This provides further enrichment performed after the filtering. *For example, you may have a simple sine wave on the input, processed through the input distortion, which adds several harmonics. A filtering using a low-pass filter may then remove most of the higher harmonics content. Saturation may then be used to generate some of the harmonics back again.*

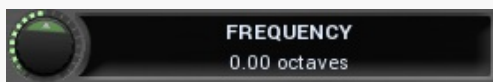
Range: 0.00% to 100.0%, default 0.00%



## Output gain

Output gain defines the output gain that is applied after the filter. This could be useful for controlling the input to the next stages - the next filter, global saturation. The rule of thumb is - the higher the **Drive** or filter **Gain**, the lower this output gain should be to compensate. *For example, you may set a high drive followed by a Sub-X with high gain in the first filter. Its input distortion will generate lots of higher harmonics as well as change the output level, and the filter would increase the level even more. This could easily be more than +20dB, which when fed to the following filter's distortion or global saturation, may be unusable as each of these nonlinear processors would be immediately overdriven. Just use the filter's output gain to compensate for this by dropping it down.*

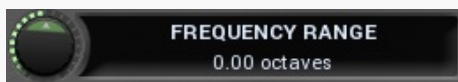
Range: -48.00 dB to +48.00 dB, default 0.00 dB



## Frequency

Frequency controls the filter minimum frequency.

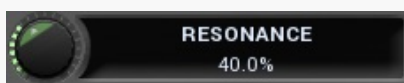
Range: -8.00 octaves to +8.00 octaves, default 0.00 octaves



## Frequency range

Frequency range controls the extent to which the frequency is modulated. With 0% no modulation occurs and the filter frequency is defined by the **Frequency** parameter only. If you increase the range however, the filter frequency will move within the range frequency ... frequency + range according to the ADSR envelope.

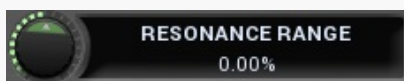
Range: -8.00 octaves to +8.00 octaves, default 0.00 octaves



## Resonance

Resonance controls the filter central resonance. Please note that it is used only for some filter types.

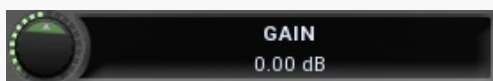
Range: 0.00% to 100.0%, default 40.0%



## Resonance range

Resonance range controls the extent to which the resonance is modulated. With 0% no modulation occurs and the filter resonance is defined by **Resonance** parameter only. If you increase the range however, the filter resonance will move away from the central resonance value according to the ADSR envelope.

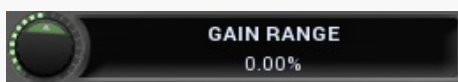
Range: -100.0% to 100.0%, default 0.00%



## Gain

Gain controls the central gain of the filter. Please note that it is used only for some filter types.

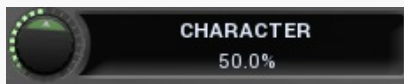
Range: -48.00 dB to +48.00 dB, default 0.00 dB



## Gain range

Gain range controls the extent to which the gain is modulated. With 0% no modulation occurs and the filter gain is defined by the **Gain** parameter only. If you increase the range however, the filter gain will move away from the central gain value according to the ADSR envelope.

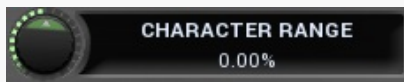
Range: -100.0% to 100.0%, default 0.00%



## Character

Character controls the central character of the filter. Please note that it is used only for some filter types. Character affects some additional filter specific features, such as dispersion of harmonics. For polymorph filters character actually controls the internal structure of the filter and any change to this value completely changes the algorithm providing maximum unique sound combinations. Therefore character modulation is not available for polymorph filters, as modulation would create extreme changes.

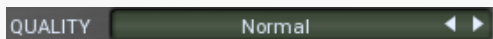
Range: 0.00% to 100.0%, default 50.0%



## Character range

Character range controls the extent to which the character is modulated. With 0% no modulation occurs and the filter character is defined by **Character** parameter only. If you increase the range however, the filter character will move away from the central character value according to the ADSR envelope.

Range: -100.0% to 100.0%, default 0.00%



## Quality

Quality controls the ratio between audio quality and CPU requirements.



## Type

Type defines the type of filter. Note that different filters may consume different amounts of CPU. By definition a filter does not produce any frequencies which are not already in the signal, hence the name "filter". The difference between the types is how each filter modifies the levels of each frequency. Some filters completely remove certain frequencies, others just change the levels of certain frequencies. If you wish to make the signal richer by creating additional frequencies which are NOT in the signal yet, use a distortion or saturation plugin.

**Low-pass, high-pass, band-pass and notch** filter out some frequencies completely. Low-pass filter, for example, lets all frequencies below a certain limit pass and removes everything above. This is possible only in theory though, so you might say that the higher the frequency is above the filter frequency, the more it is attenuated. The higher the slope is, the steeper the filter is, hence it removes more of the unwanted frequencies.

Traditional low-pass filters have a 12dB/octave slope, which means that, for example, if you have that filter set at 1kHz and the Q is configured so that at 1kHz the gain is -3dB (which is usually the default, technical reasons), then at 2kHz (+1 octave) it is -15dB, at 4kHz (+2 octaves) it is -27dB etc. Our filters can provide up to 120dB/octave slope, so it can pretty much kill everything above it within a single octave.

High-pass filter works the same way, but kills everything below its frequency. Notch kills everything at the filter frequency plus some adjacent frequency range (determined by the filter's Q value), while band-pass works the other way around - it only lets through the filter frequency and the adjacent frequency range.

**Peak and shelf** filters are similar to those used in equalizers.

**Fade** filters provide cross-fades between low-pass and high-pass filters and other combinations. Use the **Character** parameter to control how much LP and how much HP is used then.

**Harmonics** filters are complex combinations of peak filters designed to process multiple harmonics of the base frequency. Basically if you configure a harmonic filter at say 100Hz, then there will be series of peak filters at 100Hz, 200Hz, 400Hz etc. or (100Hz, 200Hz, 300hz... if the linear version is used). The **character** parameter controls the level of succeeding harmonics. For example, if character is 0%, then it is basically just an ordinary peak filter. If it is 100%, then there is one filter for all available harmonics, each with the same gain. For something in between, the gain for each higher harmonic is lower than the previous one.

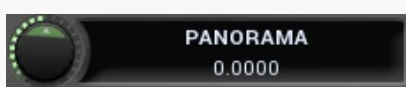
**Linear** harmonics filters affect linear multiples of the base frequency, while normal harmonics filters only affect power-2 multiples, hence octaves above the base. **Swap** versions cause inverted gain for odd and even harmonics.

**Sub-X, over-X and band-X** filters are specialized complex combinations of other filters originally designed for wobbling basses. These mainly combine LP/HP/BP filters with harmonic filters. The **character** parameter controls the distribution of harmonics and should be used simply by trial-and-error.

**Formant** filters are filters emphasizing vowel sounds. There are filters for each vowel and the newest filter, called **Formant A-E-I-O-U** cross-fades between these 5 vowels, depending on the **character** parameter. To get reasonable "talkbox" sounds, it is recommended to use a rich audio signal (e.g. saw wave).

**Comb and diffuser** filters are complex comb filtering processors with pretty wild and fat responses. These range from simple comb filtering to complex almost ambient responses. Each filter uses a different kernel, so it shall be selected by trial-and-error approach. The **character** parameter controls the internal feedback of the filter.

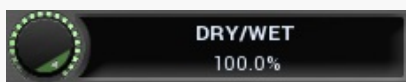
**Polymorph** filters are generic polymorphic filters, which change its internal structure according to the **Character** parameter and provide a virtually limitless number of unique sound combinations. However, these are usually also the most computationally demanding.



### Panorama

Panorama lets you shift the filter frequency between channels. Left channel's frequency is shifted down by specified amount, right channel is shifted up, third down etc.

Range: -48.0000 to +48.0000, default 0.0000



### Dry/wet

Dry/wet controls the filter dry/wet ratio. Please note that since most filters are altering phase, you may experience various frequency level changes for values other than 0% and 100%.

Range: 0.00% to 100.0%, default 100.0%

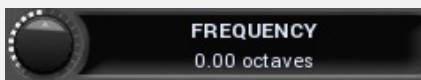
**By velocity panel**



By velocity panel lets you control how much the basic parameters are affected by the note velocity. Unlike modulators and MIDI, which are global and affect all voices when changed, these parameters lets you assign different values to each voice depending on the velocity (or aftertouch).

### AFTERTOUCH **Aftertouch button**

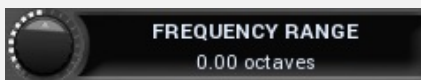
Aftertouch button enables processing aftertouch for this filter in the same way as velocity.



### **Frequency by velocity**

Frequency by velocity controls how much the filter frequency is adjusted depending on the velocity of each note. For example, if this is set to +1 octave and the filter **frequency parameter** is +2 octaves, then for a note with minimum velocity the filter frequency will be +2 octaves. If the note has maximum velocity, then the filter frequency will be +3 octaves.

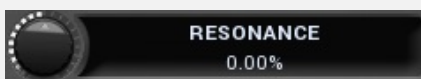
Range: -8.00 octaves to +8.00 octaves, default 0.00 octaves



### **Frequency range by velocity**

Frequency range by velocity controls how much the filter frequency range is adjusted depending on the velocity of each note. For example, if this is set to +1 octave and filter **frequency range parameter** is +2 octaves, then for a note with minimum velocity the filter frequency range will be +2 octaves. If the note has maximum velocity, then the filter frequency range will be +3 octaves.

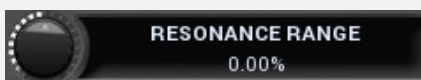
Range: -8.00 octaves to +8.00 octaves, default 0.00 octaves



### **Resonance by velocity**

Resonance by velocity controls how much the filter central resonance is adjusted depending on the velocity of each note. For example, if this is set to +30% and filter **resonance parameter** is 40%, then for a note with minimum velocity the filter resonance will be 40% of an octave. If the note has maximum velocity, then the filter resonance will be 70%.

Range: -100.0% to 100.0%, default 0.00%



### **Resonance range by velocity**

Resonance range by velocity controls how much the filter resonance range is adjusted depending on the velocity of each note. For example, if this is set to +30% and filter **resonance range parameter** is 40%, then for a note with minimum velocity the filter resonance range will be 40% of an octave. If the note has maximum velocity, then the filter resonance range will be 70%.

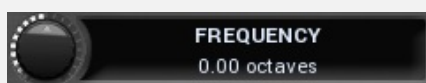
Range: -100.0% to 100.0%, default 0.00%



# By note panel



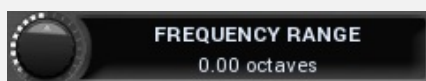
By note panel lets you control how much the basic parameters are affected by which note is being pressed. Unlike modulators and MIDI, which are global and affect all voices when changed, these parameters let you assign different values to each voice depending on the note.



## Frequency by note

Frequency by note controls how much the filter frequency is adjusted depending on which note is being pressed. For example, if this is +1 octave and filter **frequency parameter** is +2 octaves, then for the lowest note the filter frequency will be +2 octaves. For the highest note the filter frequency will be +3 octaves.

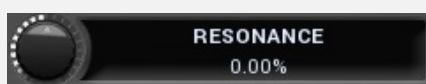
Range: -8.00 octaves to +8.00 octaves, default 0.00 octaves



## Frequency range by note

Frequency range by note controls how much the filter frequency range is adjusted depending on which note is being pressed. For example, if this is +1 octave and filter **frequency range parameter** is +2 octaves, then for the lowest note the filter frequency range will be +2 octaves. For the highest note the filter frequency range will be +3 octaves.

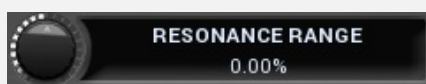
Range: -8.00 octaves to +8.00 octaves, default 0.00 octaves



## Resonance by note

Resonance by note controls how much the filter central resonance is adjusted depending on which note is being pressed. For example, if this is +30% and filter **resonance parameter** is 40%, then for the lowest note the filter resonance will be 40% of an octave. For the highest note the filter resonance will be 70%.

Range: -100.0% to 100.0%, default 0.00%



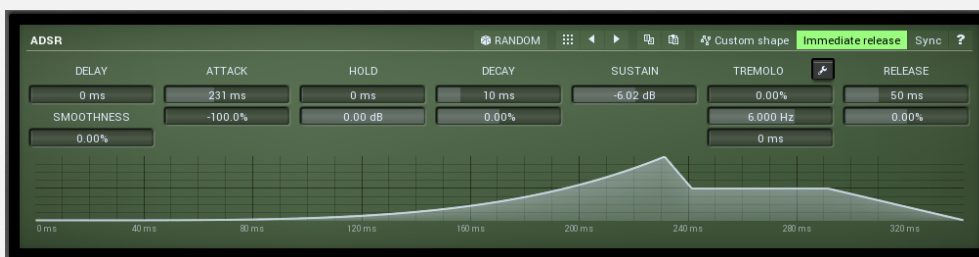
## Resonance range by note

Resonance range by note controls how much the filter resonance range is adjusted depending on which note is being pressed. For example, if this is +30% and filter **resonance range parameter** is 40%, then for the lowest note the filter resonance range will be 40% of an octave. For the highest note the filter resonance range will be 70%.

Range: -100.0% to 100.0%, default 0.00%



# ADSR graph



ADSR graph controls the ADSR envelope for the filter. Our ADSR envelopes are much more sophisticated than classic attack-release-sustain-decay envelopes. Besides these common parameters they also let you control the curvature of each stage. Additionally, there are hold and delay sections, global smoothing and tremolo. You can even use the custom shape mode to define your own attack/release curves.

## **Random button**

Random button generates random settings using the existing presets.

## **Presets button**

Presets button displays a window where you can load and manage available presets. Hold **Ctrl** when clicking to load a random preset instead.

## **Left arrow button**

Left arrow button loads the previous preset.

## **Right arrow button**

Right arrow button loads the next preset.

## **button**

This button copies the settings onto the system clipboard.

## **button**

This button loads the settings from the system clipboard.

## **Custom shape button**

Custom shape button enables custom shape mode, which lets you draw your own attack and release stages using the envelope system. Both stages are then automatically connected to form the resulting envelope.

## **Immediate release button**

Immediate release button activates the immediate release mode in which case the note-off causes an immediate switch to the release stage. If this is disabled, the release stage does not occur until the whole attack/decay stage finishes.

## **Sync button**

Sync button controls the ADSR tempo sync feature. By default this is disabled and means that all times are followed exactly, meaning that if **Attack** is say 100ms, then it will be 100ms indeed. Tempo sync lets the plugin adjust the times to ensure it will be always in sync with the host tempo. In this case 100ms may become say 125ms if the tempo is 120bpm, because 125ms is the length of a 16th note. This makes it

extremely simple to convert any envelope to a tempo-synced one. The plugin always chooses the nearest longer note, in other words it always round up.

**Straight** and **Triplets** modes automatically find 'nice' values.

*For example, if a 16th note takes 100ms, the attack time is 550ms, and the sync mode is straight, then the plugin checks for 100ms, find out that it is too low, so it checks 8th note, being 200ms, still too low, then continues with quarter note, which takes 400ms, and still not enough, finally 800ms corresponding to a half note is the one, so the resulting time will be 800ms. Triplet cases are more complex, but the principle is the same.*

**1/16**, **1/8** and **1/4** modes choose the nearest higher multiply of the base note length. For example, if a 16th note takes 100ms, the attack time is 550ms, and the sync mode is 1/16, the resulting time will be 600ms.

## Settings button

Settings button displays additional tremolo settings, containing tremolo behaviour and shape.

# FX panel



FX panel controls the effects engine. The effects are applied to all voices at once, so for example distortion in the filters reacts very differently from distortion in the effects. The modular pipeline makes dozens of ultra-high quality effects available and you can connect them together in any way that you want. The signal always flows from top to bottom, left to right through various processors. The engine can also generate complete structures, just in case you need some inspiration. The top row contains the available inputs and the bottom row shows the outputs, which are simply mixed together. You can disable any output by clicking on it (that also disconnects the processors in that chain). Click on an empty field to add a processor to it. Use **right mouse button** (or **Ctrl + click**) on a processor to get some additional settings, mostly about routing audio and

MIDI. There are also a few shortcuts: **double click** enables or disables the processor, **Alt + click** deletes the processor, and **Shift + click** displays the processor in a popup window.

The set of processors includes everything from modulation effects, delays, distortions, dynamics, filters etc., and also contains several building blocks - mixer, crossover, LR <-> MS (de)encoder, LFO, channel matrix and many more. Each processor has some inputs and outputs, in most cases it is just 1 input and 1 output as you might expect. There are also processors with side-chain inputs (e.g. Ratio or Dynamics), processors with multiple outputs (e.g. Crossover), processors with multiple inputs (e.g. Mixer) and even processors with no inputs (e.g. Oscillator, LFO). The inputs and outputs are intelligently mapped by default, in most cases the whole structure "connects", but sometimes you may need to use the **right mouse button** to get the additional routing settings.

Most processors are self-explanatory, but a few of them could use some additional information:

**Modular** processor is actually "itself", the whole modular pipeline inside itself. This is useful for example when the structure is starting to get cluttered. Note that you may run into GUI problems - it is impossible to fit that amount of controls onto a screen without huge resolution, so you may need to use the **shift + click** to get the modular processor displayed in a pop-up window.

**Feedback** processor serves as a feedback source. Normally the feedback would not be possible, because the signal always flows from top to bottom. The feedback processor lets you go around this. It has no inputs, but generates a signal anyway - by taking it from somewhere else, from previous processing to be precise. Any of the processors can "generate feedback" into one of the feedback channels. You can imagine that each processor can not only process the signal somehow and pass the results to the output, it can also store the output in one of the feedback channels, just in case some feedback processor would use it.

The feedback processor does not have many parameters, but 2 of them are very important - feedback and delay. First, feedback is the input gain. If it is too high, reaching 0dB, you may expect the typical infinite feedback screaming. The processor provides a clipper, but it will not save you, so be careful when using it. Second, delay - it is physically impossible to have zero delay feedback, not in the analog world and not in the digital world either. So the delay controls how much time it takes to get back. And here is the catch - for higher delays, say above 5ms, there are no problems, but with lower delays the CPU consumption can rise exponentially. There are technical reasons for that and it can never be improved, think about it as physical limitations.

How to use the feedback then? First, create a feedback processor somewhere, it will use the feedback channel 1 by default. Then right-click on any of the existing processors and set "generate feedback 1", that is all. If you do not have any processors there and you just want to use feedback as some kind of experimental delay generator, use some low-CPU processor, such as Utility.



## Randomize button

Randomize button (with the text 'Random') generates random settings. Generally, randomization in plug-ins works by selecting random values for all parameters, but rarely achieves satisfactory results, as the more parameters that change the more likely one will cause an unwanted effect. Our plugins employ a smart randomization engine that learns

which settings are suitable for randomization (using the existing presets) and so is much more likely to create successful changes.

In addition, there are some mouse modifiers that assist this process. The smart randomization engine is used by default if no modifier keys are held.

Holding **Ctrl** while clicking the button constrains the randomization engine so that parameters are only modified slightly rather than completely randomized. This is suitable to create small variations of existing interesting settings.

Holding **Alt** while clicking the button will force the engine to use full randomization, which sets random values for all reasonable automatable parameters. This can often result in "extreme" settings. Please note that some parameters cannot be randomized this way.

Hold **Shift** while clicking the button to undo the previous randomization.

 A-levels normalizer

### **Presets button**

Presets button displays a window where you can load and manage available presets. Hold **Ctrl** when clicking to load a random preset instead.

### **Left arrow button**

Left arrow button loads the previous preset.

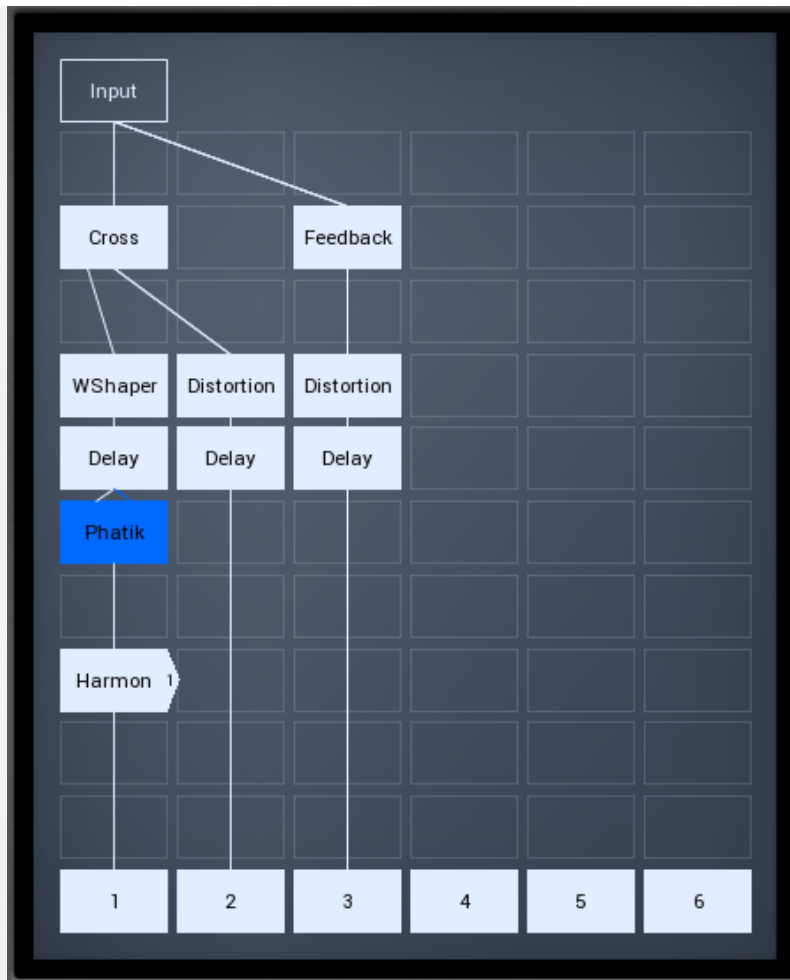
### **Right arrow button**

Right arrow button loads the next preset.

### **Randomize button**

Randomize button loads a random preset.

## **Modular editor**



Modular editor lets you edit the processing flow. There is at least one input at the top, several outputs at the bottom and several processing boxes in-between. Processing is always performed from top to bottom and is indicated by a solid line. All outputs except the first one are disabled by default, unless you put a plugin into its chain. Every output can be disabled by clicking on it.

Each processing box can contain one plugin. By default all plugins are taking their inputs directly from above (thus from the same chain that they are in), but if you for example put a plugin in the first row of the second chain, there is no input above, so it is redirected to the first input. Some plugins may have multiple input channels or even a variable number of input channels. Most plugins have just one output channel, in which case they place the output into the same chain that they are in. Some plugins have variable output channels in which case they send them to the same channel, the next one to the right and so on.

If you **click** on an empty box a menu with all available processing plugins will be shown. If you click on a box which already contains a plugin, it will be selected and its editor shown to the right. The currently-selected plugin is highlighted. If you **double-click** or click using **right mouse button** on it, the advanced settings window will be displayed. You can use it to configure the plugin input and output channels and much more. Hold **Ctrl** and click on the plugin to enable or disable it. Hold **Alt** and click on the plugin to delete it. Both actions are available from the advanced settings as too. Hold **Shift** and click on the plugin to show its editor in a pop-up window.



## Collapse button

Collapse button minimizes or enlarges the panel to release space for other editors.

# Phatik 1



## **Randomize button**

Randomize button (with the text 'Random') generates random settings. Generally, randomization in plug-ins works by selecting random values for all parameters, but rarely achieves satisfactory results, as the more parameters that change the more likely one will cause an unwanted effect. Our plugins employ a smart randomization engine that learns which settings are suitable for randomization (using the existing presets) and so is much more likely to create successful changes.

In addition, there are some mouse modifiers that assist this process. The smart randomization engine is used by default if no modifier keys are held.

Holding **Ctrl** while clicking the button constrains the randomization engine so that parameters are only modified slightly rather than completely randomized. This is suitable to create small variations of existing interesting settings.

Holding **Alt** while clicking the button will force the engine to use full randomization, which sets random values for all reasonable automatable parameters. This can often result in "extreme" settings. Please note that some parameters cannot be randomized this way.

Hold **Shift** while clicking the button to undo the previous randomization.



## Presets button

Presets button displays a window where you can load and manage available presets. Hold **Ctrl** when clicking to load a random preset instead.

## Left arrow button

Left arrow button loads the previous preset.

## Right arrow button

Right arrow button loads the next preset.

## Randomize button

Randomize button loads a random preset.

## Show window button

Show window button displays the plugin in a dedicated pop-up window. You can do the same thing by Shift + click on the processor item in the modular grid.

# Globals panel



Globals panel controls the main parameters of the plugin.

## SIDE-CHAIN Side-chain button

Side-chain button selects the side-chain input of the plugin for processing. By default it is disabled and the original input (before processing by amp, convolution and compressor) is used. Therefore the plugin makes the processed signal dynamics match the original. However you may use the side-chain input to make the processed signal match the dynamics of any other signal.

## PSYCHO-ACOUSTIC PREFILTERING Psycho-acoustic prefiltering

Psycho-acoustic prefiltering enables the loudness estimation pre-filtering processor. When disabled, the level detector reacts to the input level of the incoming signal. This is the traditional way, but it has nothing to do with human hearing, which reacts differently to different frequencies - our ears hear the different frequencies of equal loudness at different levels, being most sensitive to sounds between 2 and 5 kHz, (see the Fletcher<sup>3</sup>/<sub>4</sub>-Munson curves, which are one of many sets of equal-loudness contours for the human ear) Psycho-acoustic pre-filtering pre-processes the level detection signal in a similar way to human hearing - it attenuates those frequencies we do not hear well and amplifies frequencies that we do. That way the level detector



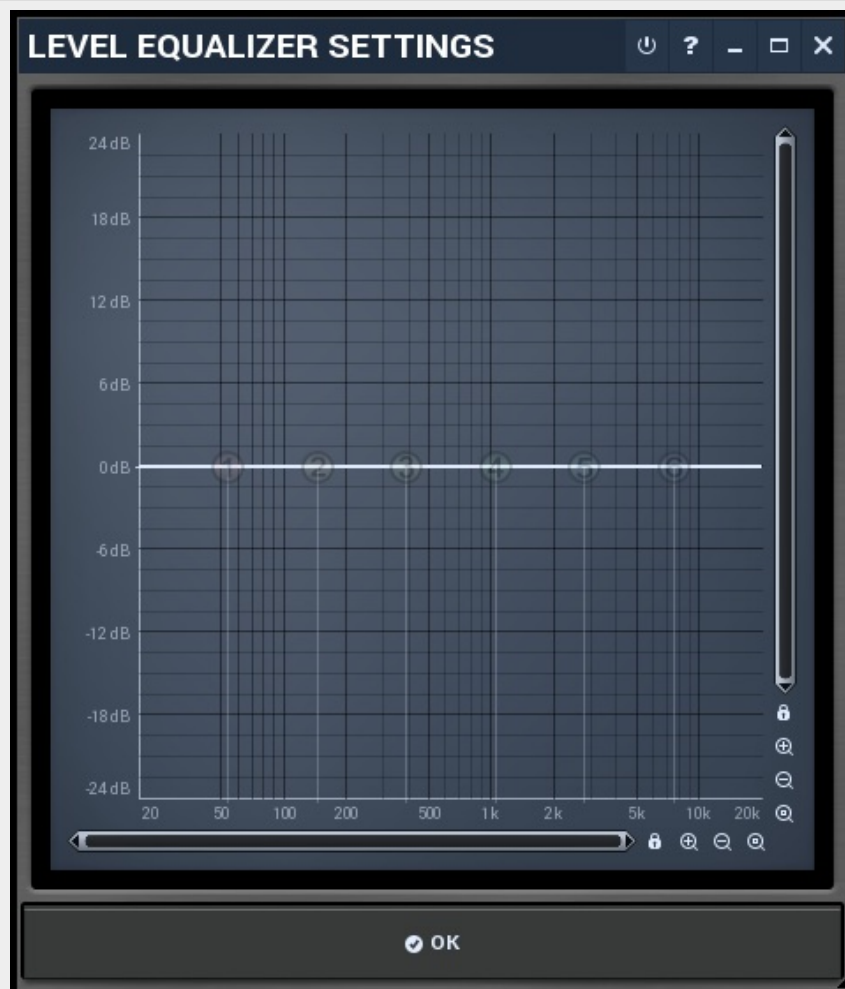
starts responding to what we actually hear, not to some sort of scientific signal as it usually does.

This feature is disabled by default simply because most users are not used to working with this feature, but it is perfectly safe to use it. However, do not use it with limiters, where you want to remove the peaks, hence you are not focussed on human hearing, but rather are dealing with the technological problems in digital and analog audio. **EQ**  
Eq button

Eq button shows the settings of the side-chain equalizer. This equalizer does not affect the outgoing signal, but processes the signal entering the level detector. You can use it to target those frequencies to which you want the processor to react. In most cases you will be using low/high/band-pass filters to remove those parts of the spectrum that you are not interested in utilizing. For example, to make the detector react to a bass drum, you may use a low-pass filter with a frequency of say 100 Hz.

Additionally, the equalizer lets you perform more complicated processing. For example, you may want the detector to react to the whole spectrum, but especially the high end of the spectrum, in which case a high-shelf filter may be the appropriate one to choose.

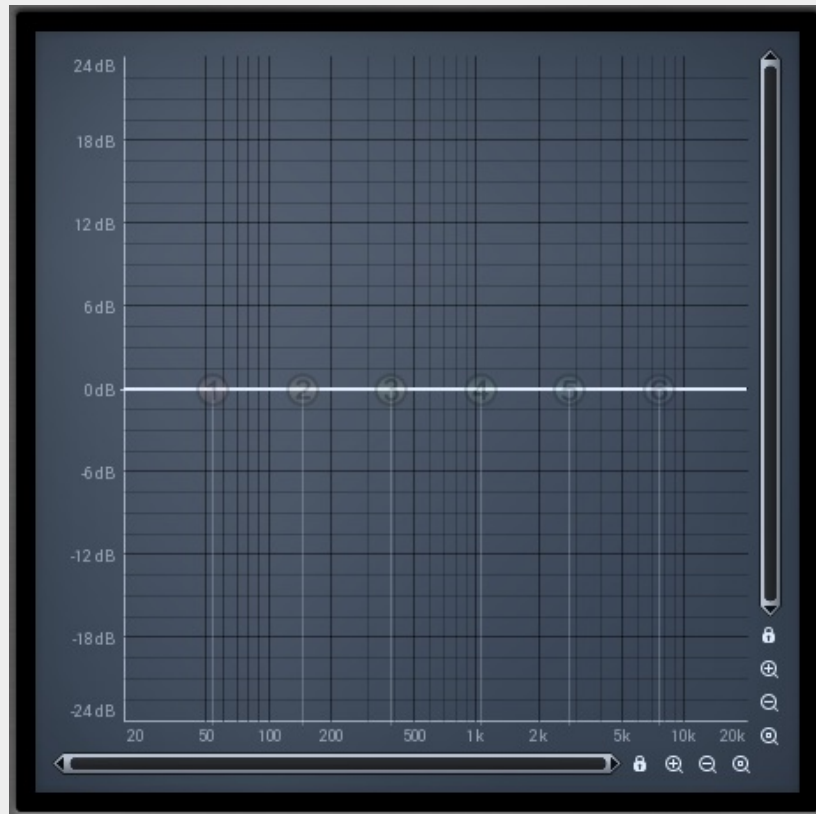
## Level equalizer settings



### button

This button enables or disables the level equalizer. It is disabled by default to lower CPU consumption.

## Equalizer shape graph



Equalizer shape graph controls and displays the frequency response. There are several bands available, each of them can be enabled/disabled, can be set to a different filter, can have different frequency, Q and other parameters.

Double-click on a band point to enable or disable a band. Drag it to change its frequency and gain. Drag the horizontal nodes to change its Q. Hold **ctrl** key for fine tuning. Click using the right mouse button on it to open a window with additional settings.

## Band settings window



Band settings window contains settings for the particular band and can be displayed by right-clicking on a band or from a band list (if provided). On the left side you can see list of available filters, click on one to select it. On the right side, additional options and features are

available.

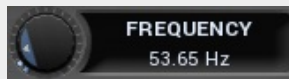
## General panel



General panel contains standard filter settings such as frequency or Q. Most of these values are available directly from the band graph, but it may be necessary to use these controls for more accurate or textual access.

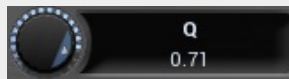
### INVERT GAIN **Invert gain**

Invert gain inverts the gain of the band, e.g. makes -6dB from +6dB.



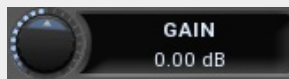
### **Frequency**

Frequency defines the band's central frequency, which has different meaning depending of filter type.



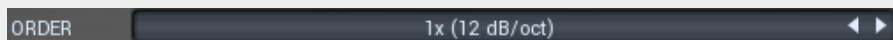
### **Q**

Q defines bandwidth. Please note that Q is an engineering term and the higher it is, the lower the bandwidth. Our implementation is trying to be more user-friendly, and by increasing the value (thus to the right), the bandwidth is increased as well. The editor still displays the Q value correctly.



### **Gain**

Gain defines how the particular frequencies are amplified or attenuated. This parameter is used only by peak and shelf filters.



### **Order**

Order can potentially duplicate some of the filters creating steeper ones. By default, the order is 12 dB/octave using 2-pole filters. By specifying 24 dB/octave the plugin uses 4-pole filters instead etc.



### **Channels**

Channels controls which channels the band processes. If the input is stereo (left and right channels), then you can make a band process only the left, only the right, or both channels. Similarly when the plugin is set to M/S channel mode, you can choose between mid, side or both channels. You cannot process left with one band and side with the other, because these are working in different encoding modes. However you can easily use 2 instances of the plugin in series, one in L/R mode and the other in M/S.

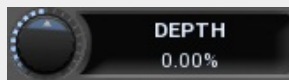
## Harmonics panel



Harmonics panel contains parameters of harmonics - cloned bands created at defined higher frequencies. This is often useful to remove natural noises, which usually bring some harmonics with them etc.

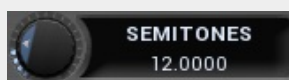
### LINEAR **Linear button**

Linear button enables the linear harmonics spacing. When the band frequency is say 100Hz, then in the default logarithmic mode the harmonics are 200Hz, 400Hz, 800Hz etc. This is suitable, because the filters themselves are logarithmic, however harmonics generated by physical instruments are not spaced in this way, but rather 200Hz, 300Hz, 400Hz, 500Hz etc. In linear mode the harmonics work this way, but note that then there is only a limited set of harmonics and Q is modified to approximate a reasonable behaviour, which is not always possible.



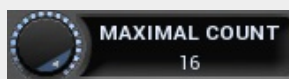
### **Depth**

Depth defines the gain of the created harmonics. With maximum depth, all harmonics will have the same gain as the original band. A lower value makes higher harmonics have lower gain. A negative depth may even make one have a positive and the other a negative gain and is particularly useful for creative effects.



### **Semitones**

Semitones defines the frequency of the harmonics. For example, if the band is at 100Hz and number of semitones is 12 (default), then the first harmonic will be at 200Hz (12 semitones higher), second at 400Hz etc., thus they are logarithmically-spaced harmonics. When linearly-spaced harmonics are enabled, this merely changes the ratio between them. In this mode, 100Hz is followed by 200Hz, 300Hz, 400Hz, 500Hz etc.



### **Maximal count**

Maximal count defines the maximum number of harmonics to be created.

## Harmonics grid

A control panel for harmonics, featuring a dark grey background with the word "HARMONICS" in white. To the right is a horizontal grid of 12 white rectangular buttons, each with a small dark square in the center, used for manually turning on or off specific harmonics.

HARMONICS

Harmonics grid is useful to turn on/off particular harmonics manually.



### Dry/wet

Dry/wet defines ratio between dry and wet signals. 100% means fully processed, 0% means no processing at all.

Range: 0.00% to 100.0%, default 100.0%



### Dynamics

Dynamics controls the amount of dynamics processing. 100% means the dynamics processor fully matches (depending on the **Mode**) the dynamics to the original, 0% means there will be no dynamics matching at all and the plugin will serve just as an amp, convolution and compression multi-effect. If you don't use the integrated effects and use the side-chain feature, then this parameter serves the same purpose as

**Dry/wet.**

Range: 0.00% to 100.0%, default 100.0%



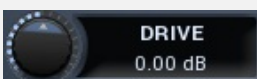
### Mode

Mode defines what kind of dynamics processing the plugin performs. Essentially the plugin measures the input level (either unprocessed input after the input gain, or the sidechain) and the output level processed by the amp, convolution and compressor and performs some operation on these levels.

**Matching** makes the output dynamics match the input dynamics. That's the most typical operation. All the included processors - amp, convolution and compressor - generally lower the output dynamics and lengthen the signal envelopes. Matching then lets you restore some of the original dynamics back.

**Matching (no boost)** works in the same way, but it never amplifies the output signal. This mode is the default.

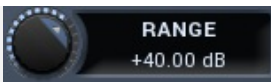
**Ducking** does the opposite - it lowers the output level if the original level increases. While this might be useful for classic side-chain ducking, it seems very useful with integrated effects as well since it produces a sort of pumping, highlighting the prolonged signal envelopes. Ducking generally requires some sort of threshold, which is defined by the **Range** parameter, it is '-range' to be exact.



### Drive

Drive defines the input gain, the power modification applied to the input signal. Please note that the input signal level (if the side-chain is not used) is measured after this input gain, hence it also affects the integrated processors, especially amp.

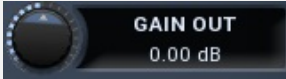
Range: -24.00 dB to +24.00 dB, default 0.00 dB



## Range

Range controls the maximum gain that the plugin can perform. Please note that its effect is not the same as dry/wet nor as the dynamics parameters, since it only ensures the gain won't exceed a particular level, but otherwise there is no change in the dynamics processing.

Range: 0.00 dB to +60.00 dB, default +40.00 dB



## Gain Out

Gain Out defines the power modification applied to the output signal. It is performed after the dynamics processing, hence its main purpose is to fix the output level so that the **Dry/wet** parameter is easier to use.

Range: -24.00 dB to +24.00 dB, default 0.00 dB



## Speed

Speed defines the time response of the plugin. It essentially controls the attack, release and other characteristics of both level followers - the first one used on the original input / side-chain and the second used on the processed signal.

Range: 1.0 ms to 1000 ms, default 40 ms



## Attack

Attack controls the time response of the plugin in addition to the **Speed** parameters and its effect usually sounds as changing the level of the initial transient, the attack. Therefore it is a very effective way to manipulate the envelope of the sound, especially useful on drums.

Range: -100.0% to +100.0%, default 0%



## Sustain

Sustain controls the time response of the plugin in addition to the **Speed** parameters and its effect usually sounds as changing the length of the sustain. While the amp, convolution and compressor usually lower the dynamics eventually making the signal envelope longer, the processor is designed to restore the original dynamics, but using this sustain parameter you can actually make the envelope longer intentionally. That's very useful for phattening drums for example.

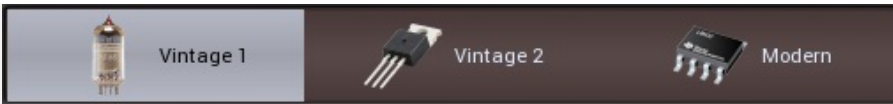
Range: -100.0% to +100.0%, default 0%

## Amp panel



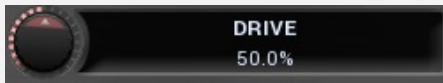
Amp panel controls the integrated vintage amp simulator. The effect order is amp -> convolution -> compressor.





## Type

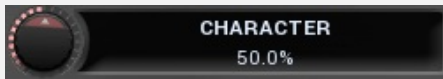
Type defines the sound character, which basically controls the amount and dirtiness of the distortion.



## Drive

Drive controls the input gain resulting in the amount of distortion.

Range: 0.00% to 100.0%, default 50.0%



## Character

Character controls the distribution of higher harmonics, hence, again, sound character.

Range: 0.00% to 100.0%, default 50.0%

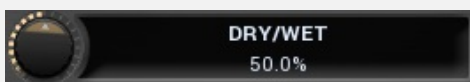
## Compressor panel



Compressor panel controls the integrated compressor. The effect order is amp -> convolution -> compressor.

### MEASURE INPUT **Measure input switch**

Measure input switch makes the compressor measure the level of the original unprocessed input. That may be handy if you use heavy distortion in amp for example, so that the compressor input already doesn't have much dynamics.

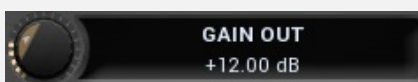


## Dry/wet

Dry/wet defines the ratio between dry and wet signals for the compressor. 100% means fully processed, 0% means no processing at all.

This feature essentially provides a modern way to do so-called parallel (or 'New York') compression. Essentially there are main 2 approaches to compression - A) set the threshold high, so that it affects everything above it, B) set the threshold low and use dry/wet to actually lower the effect of compression, which provides an easy way to control the amount of compression without too much editing of the more advanced parameters. Please note that lowering ratio does NOT have the same effect as lowering dry/wet in most cases.

Range: 0.00% to 100.0%, default 50.0%



## Gain Out

Gain Out defines the power modification applied to the output signal. Since the compressor output is followed by the main dynamics processing, this can have a big impact on the output.



Range: 0.00 dB to +40.00 dB, default +12.00 dB

## Threshold

Threshold determines the minimal signal level above which the effect is applied.

Range: -80.0 dB to 0.00 dB, default -40.0 dB

## Ratio

Ratio defines the compression ratio of the input signal above the threshold.

Range: 1.00 : 1 to Infinity, default Infinity

## Attack

Attack defines the attack time, that is how quickly the level detector increases the measured input level. Once any release time has been completed, when the input peak level is higher than the current level measured by the detector, the detector moves into the attack mode, in which the level is increased depending on the input signal. The higher the input signal, or the shorter the attack time, the faster the measured level rises. Once the measured level exceeds the **Threshold** then the dynamics processing (compression, limiting, gating) will start.

There must be a reasonable balance between attack and **release** times. If the attack is too long compared to the release, the detector will tend to keep the measured level low, because the release would cause that level to fall too quickly. In most cases you may expect the attack time to be shorter than the release time.

To understand the working of a level detector, it is best to cover the typical cases:

*In a **compressor** the attack time controls how quickly the measured level moves above the threshold and the processor begins compressing. As a result, a very short attack time will compress even the beginning transient of a snare drum for example, hence it would remove the punch. With a very long attack time the measured level may not even reach the threshold, so the compressor may not do anything.*

*In a **limiter** the attack becomes a very sensitive control, defining how much of the signal is limited and how much of it becomes saturated/clipped. If the attack time is very short, limiting starts very quickly and the limiter catches most peaks itself and reduces them, providing lower distortion, but can cause pumping. On the other hand, a higher attack setting (typically above 1ms) will let most peaks through the limiter to the subsequent in-built clipper or saturator, which causes more distortion of the initial transient, but less pumping.*

*In a **gate** the situation is similar to a compressor - the attack time controls how quickly the measured level can rise above the threshold at which point the gate opens. In this case you will usually need very low attack times, so that the gate reacts quickly enough. The inevitable distortion can then be avoided using look-ahead and hold parameters.*

*In a modulator, the detector is driving other parameters, a filter cut-off frequency for example, and the situation really depends on the target. If you want the detector to react quickly on the input level rising, use a shorter attack time; if you want it to follow the flow of the input signal slowly, use longer attack and release times.*

Range: 0 ms to 1000 ms, default 10 ms

## Release

Release defines the release time, that is how quickly the level detector decreases the measured input level. The shorter the release time, the faster the response is. Once the attack stage has been completed, when the input peak level is lower than the current level measured by the detector, the detector moves into the release mode, in which the measured level is decreased depending on the input signal. The lower the input signal, or the shorter the release time, the faster the measured level drops. Once the measured level falls under the **Threshold** then the dynamics processing (compression, limiting, gating) will stop.

There must be a reasonable balance between **attack** and release times. If the attack is too long compared to release, the detector would tend to keep the level low, because release would cause the level to fall too quickly. Hence in most cases you may expect the attack time to be shorter than the release time.

To understand the working of a level detector, it is best to cover the typical cases:

*In a **compressor** the release time controls how quickly the measured level falls below the threshold and the compression stops. As a result a very short release time makes the compressor stop quickly, for example, leaving the sustain of a snare drum intact. On the other hand, a very long release keeps the compression working longer, hence it is useful to stabilize the levels.*

*In a **limiter** the release time keeps the measured level above the limiter threshold causing the gain reduction. Having a very long release time in this case doesn't make sense as the limiter would be working continuously and the effect would be more or less the same as simply decreasing the input gain manually. However too short a release time lets the limiter stop too quickly, which usually causes distortion as the peaks through the limiter to the subsequent in-built clipper or saturator. Hence release time is used to avoid distortion at the expense of decreasing the output level.*

*In a **gate** the situation is similar to a compressor - the release time controls how quickly the measured level can fall below the threshold at which point the gate closes. Having a longer release time in a gate is a perfectly acceptable option. The release time will basically control how much of the sound's sustain will pass.*

*In a modulator, the detector is driving other parameters, a filter cut-off frequency for example, and the situation really depends on the target. If you want the detector to react quickly on the input level falling, use a shorter release time; if you want it to follow the flow of the input signal slowly, use longer attack and release times.*

Range: 1.0 ms to 10000 ms, default 100 ms

## Convolution panel



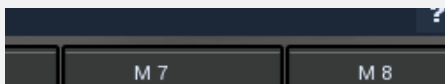
Convolution panel controls the integrated convolution reverb. The effect order is amp -> convolution -> compressor.



### **Dry/wet**

Dry/wet defines ratio between dry and wet signals. 100% means fully processed, 0% means no processing at all.

Range: 0.00% to 100.0%, default 50.0%



### **Widening**

Widening defines the broad-band stereo field widening depth. The algorithm is fully mono-compatible as it only extends the existing stereo field and no new signal is added. This parameter should only be used to control the existing stereo field.

Widening converts the audio into its mid (mono) and side channels, leaving the mid intact and applying a gain to the side channel, then converts the signal back to left and right channels. As a result the stereo image becomes wider (for widening above 0%) or narrower (for widening below 0%). This method of widening the stereo image may initially sound pleasing, however it can quickly become fatiguing on the ear and often sounds unnatural, especially for larger amounts of widening. Use this parameter to control the existing stereo field and as a special effect. Use it to increase width only with caution.

Range: Mono to 200.0%, default 0.00%

## **File**



File lets you choose an impulse response file to process. The Factory library installed with the plugin comes with lots of IRs from plates to rooms, halls, box simulations etc. Plates are often a good choice to start with.

## Favourites button

Favourites button lets you manage your personal favourites. Since you may have several file libraries installed on your system, it may be clumsy to search for them every time. Instead you can simply store them in favourites and then quickly go to the particular location without exploring the whole system.

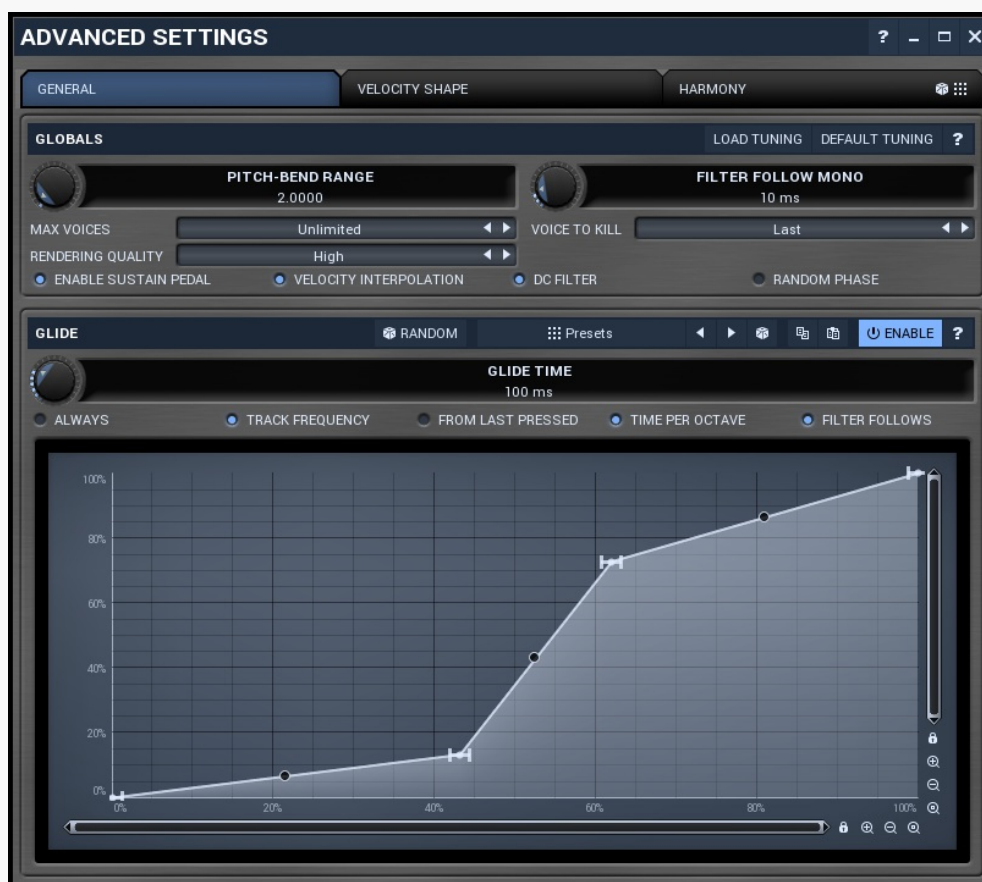
ADVANCED SETTINGS

**Advanced**

## settings button

Advanced settings button displays additional settings.

# Advanced settings window



Advanced settings window contains more advanced settings, which are used less often and so are intentionally not shown on the main plugin editor.

GENERAL

VELOCITY SHAPE

HARMONY

**Tab selector**

Tab selector switches between subsections.

## **Randomize button**

Randomize button generates random settings for the tab.

## **Presets button**

Presets button chooses a random preset for the tab.

# Globals panel



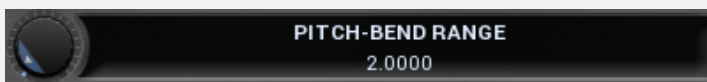
Globals panel contains some global settings controlling the plugin behaviour.

### **LOAD TUNING** **Load tuning button**

Load tuning button lets you load TUN files containing custom micro-tuning, which will replace the default equal temperament tuning (in which the logarithmic distance between every 2 semitones is exactly the same).

### **DEFAULT TUNING** **Default tuning button**

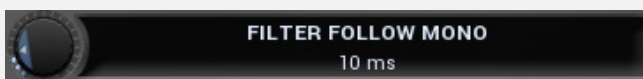
Default tuning button restores the default equal temperament tuning.



### **Pitch-bend range**

Pitch-bend range controls the range in semitones of the global pitch change caused by the MIDI pitch-bend controller.

Range: 0.000000 to 24.000000, default 2.000000



### **Filter follow mono**

Filter follow mono controls the time it takes for the filter to follow note changes in monophonic modes. Use 0ms to make the filter follow instantly, which can cause short glitches. Higher times may provide additional expression, because the filter frequency will highly depend on the first note played.

Range: 0 ms to 10000 ms, default 10 ms



### **Max voices**

Max voices controls the maximum number of voices that can play at any moment. You can use this to avoid overloading CPU when you play too many voices.



### **Voice to kill**

Voice to kill defines which voice is removed, when the number of voices exceeds the limit controlled by **Max voices** parameter.



### **Rendering**

Rendering controls the quality for rendering. If it is better than the **Quality** parameter, it will be used when rendering your project offline (if your host informs the

plugin about it). You can use this option if you want to save CPU power, but still want to maximize the audio quality for the final rendering. Be aware that when different settings are used, the output may sound significantly different.

• ENABLE SUSTAIN PEDAL

## Enable sustain pedal

Enable sustain pedal makes the synthesizer listen to the sustain pedal when processing notes. If this is disabled, the synth ignores the sustain pedal and makes only note-on/off messages relevant.

• VELOCITY INTERPOLATION

## Velocity interpolation

Velocity interpolation activates velocity interpolation for monophonic modes. When this is disabled, only the first note's velocity is relevant and the remaining notes will have the same velocity. If this is enabled however, the velocities of all notes are relevant and a quick transition between them will occur.

• DC FILTER

## DC filter

DC filter activates the global DC filter. It will remove content below 20Hz, which isn't audible but is interfering with many processing algorithms. Please note, that when the DC filter is used, the rendered oscillator shapes may start looking significantly different - square waves won't look like square waves at all. This is not an error, the generators will still sound the same, just the content below 20Hz will be removed, which is what causes the signal shape difference.

• RANDOM PHASE

## Random phase

Random phase activates the random initial phase mode for all oscillators, which kind of simulates the behaviour of analog synthesizers, where each note is just slightly different. Please note that if **Unison** is enabled, this option will be ignored and the unison's settings will be used.

# Glide panel



Glide panel contains parameters of the glide feature, which lets new notes glide from



previous ones.

 RANDOM

## Random button

Random button generates random settings using the existing presets.

 Presets

## Presets button

Presets button displays a window where you can load and manage available presets. Hold **Ctrl** when clicking to load a random preset instead.

## Left arrow button

Left arrow button loads the previous preset.

## Right arrow button

Right arrow button loads the next preset.

## Randomize button

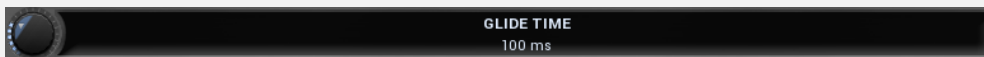
Randomize button loads a random preset.

## button

This button copies the settings onto the system clipboard.

## button

This button loads the settings from the system clipboard.



## Glide time

Glide time controls the length of the glide.

Range: 1.0 ms to 10000 ms, default 100 ms

ALWAYS

## Always

Always starts the glide every time you press a note. If this is disabled, the glide is performed only if you currently hold another note, hence the glide is started only if the 2 notes are overlapping.

*For example, if you hold key C and press key G, a glide is always performed. But if you press C, release it and then press G, the glide is started only if this parameter is enabled.*

TRACK FREQUENCY

## Track frequency

Track frequency overrides the behaviour when the notes are followed by the glide and instead it keeps track of the current frequency, so it 'starts where the last one ended'. This is useful for long glides. It even works with polyphonic glides, where at any moment the sound can split from the same pitch etc.

*For example, if the glide length is 10 seconds and you press C0 and C4 afterwards for a short time, the glide doesn't have enough time to get to C4. Normally next time the glide would start directly at C4 even though it didn't manage to get to that pitch. With this option enabled, the glide will continue where it finished (a frequency somewhere between C0 and C4).*

FROM LAST PRESSED

## From last pressed

From last pressed makes the glide start from the last key pressed, which may actually not be the one you are holding right now.

*For example, if you hold key C, then press and release key E (so there's a short glide*



from C towards E) and then press key G, there are 2 options. First, the glide may go from C to G as C is currently pressed, and that's what happens if 'from last pressed' is disabled. Second, the glide may go from E to G, because you most recently pressed E, and that's what happens with this option enabled.

• TIME PER OCTAVE

## Time per octave

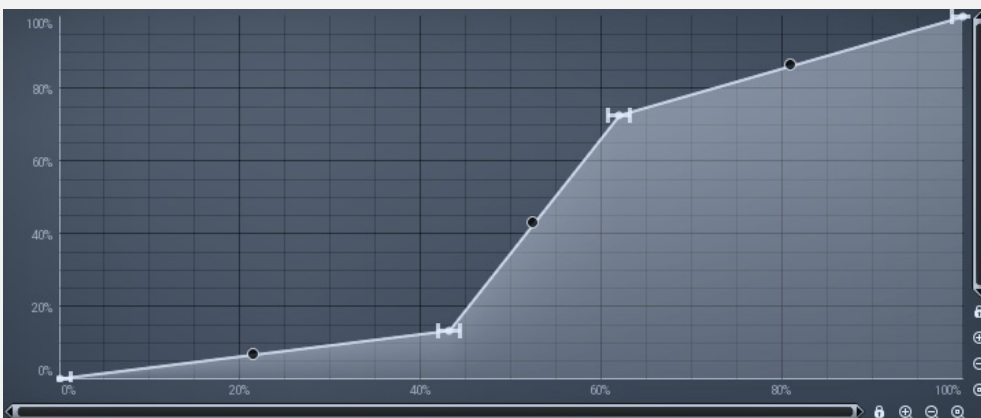
Time per octave makes the glide time depend on the pitch distance (interval) between the 2 notes. When enabled, the glide time specifies the time needed to glide across one octave. If not enabled, the glide time is the total time to get from the initial note to the target. In that case the actual glide speed differs depending on the 2 notes, because it takes the same amount of time to glide between any 2 notes, no matter if they are 1 semitone or 4 octaves away from each other.

• FILTER FOLLOWS

## FromFilter follows

FromFilter follows makes the filters follow the glide pitch. If this is disabled, the filters use the frequencies of the target notes and ignore the glide, which can be useful for example when you want the note to glide towards the target note, which will resonate using the filter.

## GlideShape



**Graph**

## editor

Graph editor lets you edit the envelope graph.

## Envelope graph

Envelope graph provides an extremely advanced way to edit any kind of shape that you can imagine. An envelope has a potentially unlimited number of points, connected

by several types of curves with adjustable curvature (drag the dot in the middle of each arc) and the surroundings of each point can also be automatically smoothed using the smoothness (horizontal pull rod) control. You can also literally draw the shape in drawing mode (available via the main context menu).

- **Left mouse button** can be used to select points. If there is a *point*, you can move it (or the entire selection) by dragging it. If there is a *curvature circle*, you can set up its tension by dragging it. If there is a *line*, you can drag both edge points of it. If there is a *smoothing controller*, you can drag its size. Hold **Shift** to drag more precisely. Hold **Ctrl** to create a new point and to remove any points above or below.
- **Left mouse button double click** can be used to create a new point. If there is a *point*, it will be removed instead. If there is a *curvature circle*, zero tension will be set. If there is a *smoothing controller*, zero size will be set.
- **Right mouse button** shows a context menu relevant to the object under the cursor or to the entire selection. Hold **Ctrl** to create or remove any points above or below.
- **Middle mouse button** drag creates a new point and removes any points above or below. It is the same as holding Ctrl and dragging using left mouse button.
- **Mouse wheel** over a point modifies its smoothing controller. If no point is selected, then all points are modified.
- **Ctrl+A** selects all points. **Delete** deletes all selected points.

## Velocity shape



### Presets button

Presets button displays a window where you can load and manage available presets.

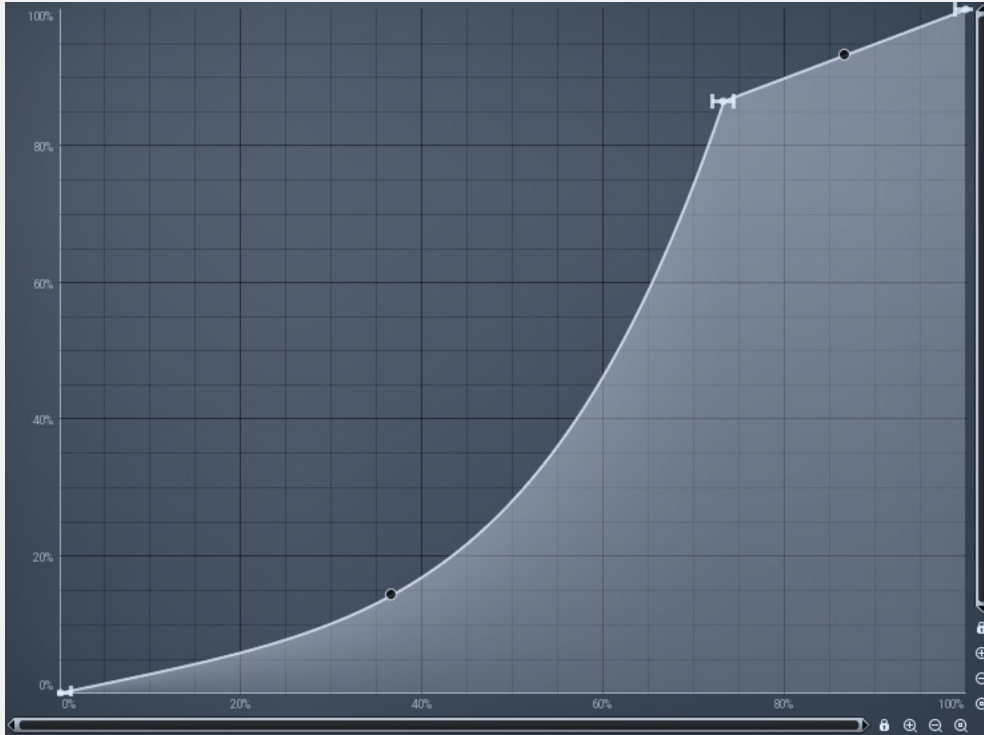
Hold **Ctrl** when clicking to load a random preset instead.

## ◀ Left arrow button

Left arrow button loads the previous preset.

## ▶ Right arrow button

Right arrow button loads the next preset.



**Graph**

## editor

Graph editor lets you edit the envelope graph.

# Envelope graph

Envelope graph provides an extremely advanced way to edit any kind of shape that you can imagine. An envelope has a potentially unlimited number of points, connected by several types of curves with adjustable curvature (drag the dot in the middle of each arc) and the surroundings of each point can also be automatically smoothed using the smoothness (horizontal pull rod) control. You can also literally draw the shape in drawing mode (available via the main context menu).

- **Left mouse button** can be used to select points. If there is a *point*, you can move it (or the entire selection) by dragging it. If there is a *curvature circle*, you can set up its tension by dragging it. If there is a *line*, you can drag both edge points of it. If there is a *smoothing controller*, you can drag its size. Hold **Shift** to drag more precisely. Hold **Ctrl** to create a new point and to remove any points above or below.
- **Left mouse button double click** can be used to create a new point. If there is a *point*, it will be removed instead. If there is a *curvature circle*, zero tension will be set. If there is a *smoothing controller*, zero size will be set.
- **Right mouse button** shows a context menu relevant to the object under the cursor or to the entire selection. Hold **Ctrl** to create or remove any points above or below.
- **Middle mouse button** drag creates a new point and removes any points above or

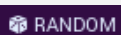
below. It is the same as holding Ctrl and dragging using left mouse button.

- **Mouse wheel** over a point modifies its smoothing controller. If no point is selected, then all points are modified.
- **Ctrl+A** selects all points. **Delete** deletes all selected points.

## Harmony panel

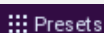


Harmony panel controls additional voices created virtually by the synth. When you press a note, the plugin may behave as if you had pressed additional notes with different velocities, tuning, delays etc.



### Random button

Random button generates random settings using the existing presets.



### Presets button

Presets button displays a window where you can load and manage available presets. Hold **Ctrl** when clicking to load a random preset instead.

### Left arrow button

Left arrow button loads the previous preset.

### Right arrow button

Right arrow button loads the next preset.

### Randomize button

Randomize button loads a random preset.

## **button**

This button copies the settings onto the system clipboard.

## **button**

This button loads the settings from the system clipboard.

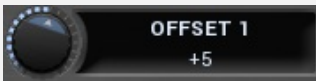
## **LEARN** **Learn button**

Learn button lets you learn the whole harmony by playing it. Enable the button, then press a chord on your keyboard and the plugin will configure the harmony settings so that the plugin will play the chord when one key is pressed.

## Note panel



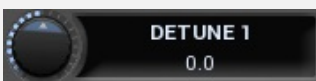
Note panel contains parameters of a virtual note. If you enable the note and press a key, the synth will behave as if you had pressed another key at the same time, with different velocity, tuning, delay etc. Note that the keys cannot be duplicates, so if you for example hold C1 and there is a virtual note 12 semitones up (C2) both are played, but when you then press C2 as well, this one won't generate a new sound and will be ignored as it is already playing, but C3 will be added as well.



### **Offset**

Offset controls the offset of the virtual note from the original one in semitones.

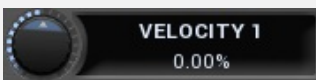
Range: -48 to +48, default +5



### **Detune**

Detune controls the detuning of the virtual note in cents. The actual pitch difference is the sum of these two control values.

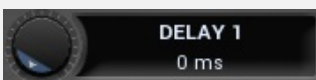
Range: -100.0 to +100.0, default 0.0



### **Velocity**

Velocity controls the difference in velocity of the virtual note from the original one.

Range: -100.0% to 100.0%, default 0.00%



### **Delay**

Delay makes the virtual note delayed from the original one and can be used for strumming effects for example.

Range: 0 ms to 1000 ms, default 0 ms

# Global meter view



Global meter view provides a powerful metering system. If you do not see it in the plug-in, click the **Meters** or **Meters & Subsystems** button to the right of the main controls. The display can work as either a classical level indicator or, in time graph mode, show one or more values in time. Use the first button to the left of the display to switch between the 2 modes and to control additional settings, including pause, disable and pop up the display into a floating window. The meter always shows the actual channels being processed, thus in M/S mode, it shows mid and side channels. In the classical level indicators mode each of the meters also shows the recent maximum value. Click on any one of these values boxes to reset them all.

**Out meter** indicates the total output level. The output meter is the last item in the processing chain (except potential downsampling and other post-processing). It is always recommended to keep the output under 0dB.

As the levels approach 0dB, that part of the meters is displayed with **red** bars. And recent peak levels are indicated by single bars.

**Width meter** shows the stereo width at the output stage. This meter requires at least 2 channels and therefore does not work in mono mode. Stereo width meter basically shows the difference between the mid and side channels.

When the value is **0%**, the output is monophonic. From 0% to 66% there is a green range, where most audio materials should remain.

**From 66% to 100%** the audio is very stereophonic and the phase coherence may start causing problems. This range is colored blue. You may still want to use this range for wide materials, such as background pads. It is pretty common for mastered tracks to lie on the edge of green and blue zones.

**Above 100%** the side signal exceeds the mid signal, therefore it is too monophonic or the signal is out of phase. This is marked using red color. In this case you should consider rotating the phase of the left or right channels or lowering the side signal, otherwise the audio will be highly mono-incompatible and can cause fatigue even when played back in stereo.

For most audio sources the width is fluctuating quickly, so the meter shows a 400ms average. It also shows the temporary maximum above it as a single coloured bar.

If you right click on the meter, you can enable/disable loudness pre-filtering, which uses EBU standard filters to simulate human perception. This may be useful to get a more realistic idea about stereo width. However, since humans perceive the bass spectrum as lower than the treble, this may hide phase problems in that bass spectrum.



**button**

This button switches between the metering view and the time-graphs. The metering view provides an immediate view of the current values including a text representation. The time-graphs provide the same information over a period of time. Since different time-graphs often need different units, only the most important units are provided.



**button**

This button increases the time-graph speed (reduces the period that is displayed).



**button**

This button decreases the time-graph speed (increases the period that is displayed).



## **button**

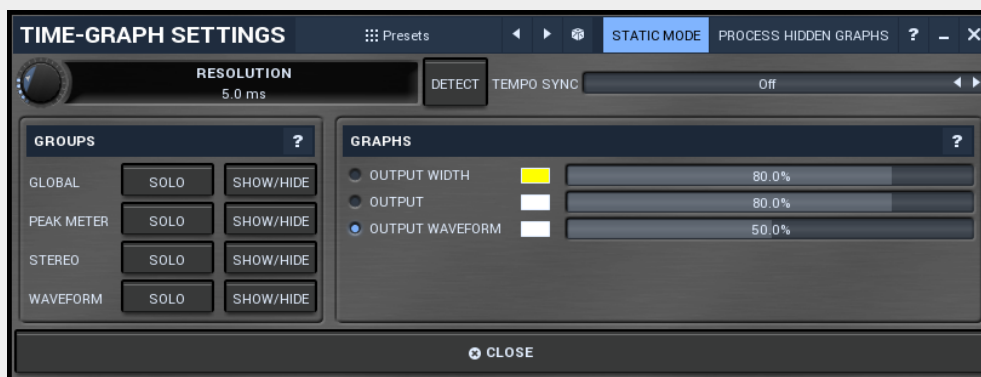
This button enables or disables the time-graph static mode. In static mode the graphs are fixed and the current position cycles from left to right; otherwise the graphs move from right to left and the current position is fixed (at the right-hand side).



## **button**

This button displays the time-graph settings. In this window you can control which graphs are displayed, the speed and other relevant parameters.

# Time-graph settings



 Presets

## **Presets button**

Presets button displays a window where you can load and manage available presets. Hold **Ctrl** when clicking to load a random preset instead.

## **Left arrow button**

Left arrow button loads the previous preset.

## **Right arrow button**

Right arrow button loads the next preset.

## **Randomize button**

Randomize button loads a random preset.

**STATIC MODE**

## **Static mode**

Static mode stops the graph from scrolling to the left and makes the graph refresh

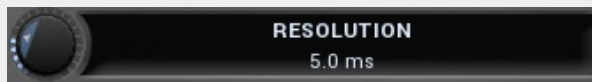
from left to right instead.

When this is disabled, the entire graph is moving from right to left as the incoming audio is processed. This may make it hard to spot the actual details, which is where the static mode comes to the rescue. Static mode is the default state and in most cases is more practical.

PROCESS HIDDEN GRAPHS

## Process hidden graphs

Process hidden graphs enables measurement of graphs which are actually disabled in the view. This may come handy if you need to repeatedly show and hide several graphs. With this mode disabled, which it is by default, the processor saves CPU resources by computing only those measurements that are actually visible. However, when you show a currently hidden graph, no measurements are available, so you will need to wait for the graph to be generated from the incoming signal. If you enable this option, the graph will be available immediately after you make it visible.



## Resolution

Resolution controls the time it takes for the graph to move one pixel. Therefore this actually controls the display speed.

DETECT

## Detect

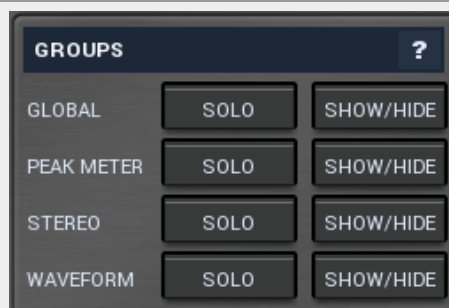
Detect enables automatic pitch detection, which lets the plugin detect and update the **Resolution** on the fly. This basically transforms the analyzer into an oscilloscope as you can see individual periods of the signal. However note that since the time-graphs show integral meters and are not designed to be an oscilloscope, it doesn't work perfectly, mainly as one period will rarely fit the screen and the whole screen will most likely be slightly moving especially for higher frequencies.



## Tempo sync

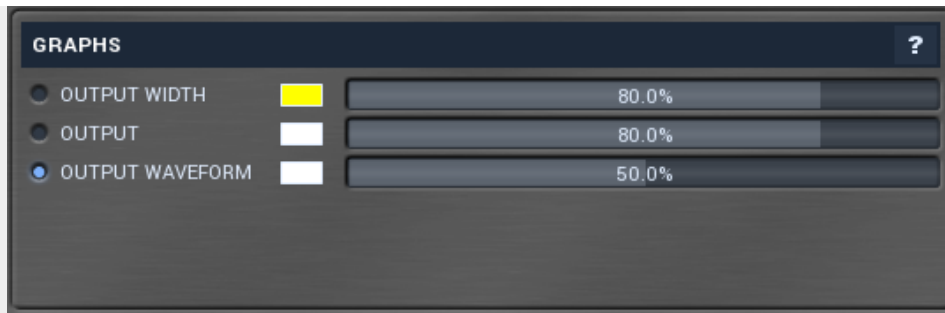
Tempo sync lets you synchronize the time-graphs to the current tempo. This way the graphs will be repeating at the selected synchronization rate.

## Groups panel



Groups panel lets you show or hide multiple graphs at once using the provided groups.

## Graphs panel



Graphs panel contains all available graphs and lets you show or hide each of them, and change their visual properties.

### **button**

This button pauses the processing.

### **button**

This button shows a pop-up window and moves the whole metering / time-graph system into it. This is especially useful in cases where you cannot enlarge the meters within the main window or such a task is too complicated. The pop-up window can be arbitrarily resized. In metering mode it is useful for easier reading from a distance for example. In time-graph mode it is useful for getting higher accuracy and a longer time perspective.

### **button**

This button enables or disables the metering system. You can disable it to save system resources.



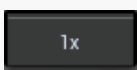
### **Collapse button**

Collapse button minimizes or enlarges the panel to release space for other editors.

# Plugin toolbar



Plugin toolbar provides some global features, A-H presets and more.



## Upsampling

Upsampling can potentially improve sound quality by processing at a higher sample rate. Processors such as compressors, saturators, distortions etc., which employ nonlinear processing generate higher harmonics of the existing frequencies. If these frequencies exceed the Nyquist rate, which equals half of the sampling rate, they get mirrored back under the Nyquist rate. This is known as aliasing and is almost always considered an artifact. This is because the mirrored frequencies are no longer harmonic and sound as digital noise as this effect does not physically occur in nature. Upsampling (or oversampling) reduces the problem by temporarily increasing the sampling rate. This moves the Nyquist frequency which in turn, diminishes the level of the aliased harmonics. Note that the point of upsampling is not to remove harmonics, we usually add them

intentionally to make the signal richer, but to reduce or attenuate the harmonics with frequencies so high, that they just cannot be represented within the sampling rate.

*To understand aliasing, try this experiment: Set the sampling rate in your host to 44100 Hz. Open MOscillator and select a "rectangle" or "full saw" waveform. These simple waveforms have lots of harmonics and without upsampling even they become highly aliased. Now select 16x upsampling and listen to the difference. If you again select 1x upsampling, you can hear that the audio signal gets extensively "dirty". If you use an analyzer (MAnalyzer or MEqualizer for example), you will clearly see how, without upsampling, the plugin generates lots of inharmonic frequencies, some of them which are even below the fundamental frequency. Here is another, very extreme example to demonstrate the result of aliasing. Choose a "sine" shape and activate 16x upsampling. Now use a distortion or some saturation to process the signal. It is very probable that you will be able to hear (or at least see in the analyzer) the aliased frequencies.*

The plugin implements a high-quality upsampling algorithm, which essentially works like this: First the audio material is upsampled to a higher sampling rate using a very complicated filter. It is then processed by the plugin. Further filtering is performed in order to remove any frequencies above the Nyquist rate to prevent aliasing from occurring, and then the audio gets downsampled to the original sampling rate.

**Upsampling also has several disadvantages of which you should be aware before you start using it.** Firstly, upsampled processing induces latency (at least in high-quality mode, although you can select low-quality mode in the plugin settings), which is not very usable in real time applications. Secondly, upsampling also takes much more CPU power, due to both the processing being performed at a higher sampling rate (for 16x upsampling at 44100 Hz, this equates to 706 kHz!), and the complex filtering. Finally, and most importantly, upsampling creates some artifacts of its own and for some algorithms processing at higher sampling rates can actually lower the audio quality, or at least change the sound character. Your ears should always be the final judge.

As always, use this feature ONLY if you can actually hear the difference. It is a common misconception that upsampling is a miraculous cure all that makes your audio sound better. That is absolutely not the case. Ideally, you should work in a higher sampling rate (96kHz is almost always enough), while limiting the use of upsampling to some heavily distorting processors.



## **Set button**

Set button uses the AGC (automatic gain compensation) processor to calculate the ideal output gain to ensure that the output audio loudness is equal to the input level. To use it, simply enable playback in your host and click the button. The plugin's output gain will be adjusted to match the input and output levels as closely as possible.

If the AGC is already enabled, the change will be instant and you can disable the AGC afterwards. Typically you will browse presets, generate random settings etc. During the entire time you will have AGC enabled to prevent you from experiencing different output loudness levels. When you find a sonically ideal setup, you simply click the Set button to set the output gain automatically and disable the AGC as you won't need it anymore.

If the AGC is not already enabled, clicking the Set button displays a window with progress bar for a few seconds, while the plugin temporarily enables AGC and analyses input and output of the plugin. After that the AGC is disabled again.

To get the best results, you should feed the plugin with some "universal" signal. If you are

processing a specific instrument, play a typical part, a chorus in case of vocals for example. If you are creating presets designed for general use, white/pink noise may be the best signal to use.

LIMITER

## Limiter button

Limiter button enables or disables the safety limiter. Its purpose is to protect you from peaks above 0dB, which can have damaging effects to your processing chain, your monitors and even your hearing.

It is generally advised to keep your audio below 0dB at all times in all stages of your processing chain. However, several plugins may cause high level outputs with certain settings, often due to unprevented resonances with specific audio materials. The safety limiter prevents that.

Note that it is NOT wise to enable this "just in case". As with any processing, the limiter requires additional processing power and modifies the output signal. It is a transparent single-band brickwall limiter, but you still need to be careful when using it.



## A-H presets selector

A-H presets selector controls the current A-H preset. This allows the plugin to store up to 8 sets of settings, including those parameters that cannot be automated or modulated. However it does not include channel mode, upsampling and potentially some other global controls available from the Settings/Settings menu.

For example, this feature can be used to keep multiple settings, when you are not sure about the ideal configuration. When you change any parameter, only the currently selected preset is modified.

The four buttons below enable you to switch between the last 2 selected sets using the A/B button, morph between the first 4 sets using the morphing button and copy & paste settings from one preset to another (via the clipboard).

It is also possible to switch between the presets using MIDI program change messages sent from your host. The set selected depends on the Program Change number: 0 selects A, 7 selects H, 8 selects A, 15 selects H and so on.



### **A/B button**

A/B button switches between the active and previously active A-H preset (not necessarily the A and B presets themselves). To compare any 2 of the A-H presets, select one and then the other. Clicking this button will then switch between these two. You can do the same thing by clicking on the particular presets, but this makes it easier, letting you close your eyes and just listen.



### **Morph button**

Morph button lets you morph between the A, B, C and D settings. Morphing only affects those parameters that can be automated or modulated; that does include most of the parameters however. When you click this button, an X/Y graph is shown allowing you to drag the position indicator to any position between the letters A, B, C and D. The closer you drag the indicator to letter 'A' for example, the closer the actual settings are to preset A.

**Please note that this will overwrite and change the preset that is currently selected, so it is best to select a new preset e.g. 'E', then use the morphing method. This way you will define the settings for A, B,C and D, morph between them, and store the result in 'E' without any modification of the original A, B, C and D presets.**

The ABCD morphing itself cannot be automated, instead the underlying parameters of the plugin can be automated. ??? Therefore if one of the ABCD presets is changed after morphing has been automated, it will make no difference to the recorded automation.



### **Copy button**

Copy button copies the current settings to the system clipboard. Other presets and upsampling settings are not copied.



### **Paste button**

Paste button pastes settings from the system clipboard into the current preset.



### **Undo button**

Undo button reverts the last change. Only changes to automatable or modulatable parameters and global settings (load/randomize) are stored.



### **Redo button**

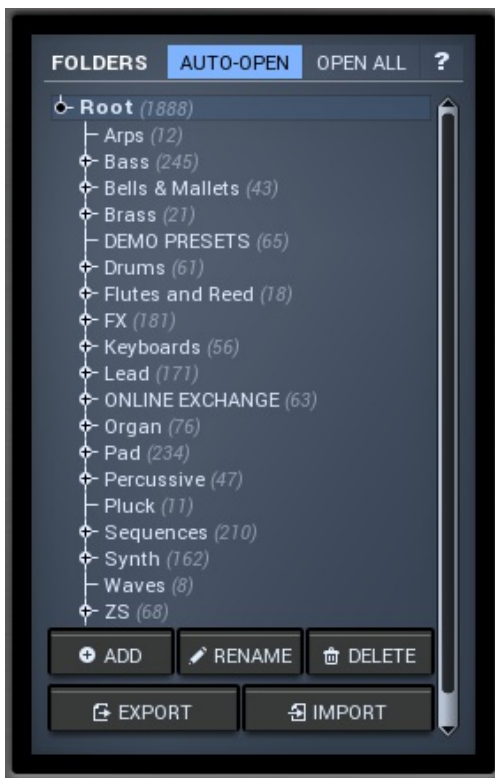


Redo button reverts the last undo operation.

# Preset management window



Preset management window provides management for your presets.



## Folders tree

Folders tree lets you organize your presets into any number of folders. Use the buttons at the bottom of the window to create, rename or delete sub-folders. Note that these are not actual files & folders on disk, but are records in the preset database.

### AUTO-OPEN **Auto-open switch**

Auto-open switch makes the tree automatically open selected items, so that all sub-folders are visible, whenever you select one. This makes it easier to browse through large structures containing many folders. The switch also makes the browser show all presets available in the selected folder including all sub-folders (except when you select the root folder).

### OPEN ALL **Open all button**

Open all button expands the whole tree, so you can see all of the folders. This may be handy when editing large preset structures.

### + ADD **Add button**

Add button creates a new folder in the tree

### ✎ RENAME **Rename button**

Rename button lets you rename the selected folder.

### 🗑️ DELETE **Delete button**

Delete button deletes the folder including all the presets and subfolders in it.

### 📁 EXPORT **Export button**

Export button lets you export the selected folder including all presets and sub-folders into a file, which you can then transfer to any computer. Or just use as a back-up.

### 📁 IMPORT **Import button**

Import button lets you import a file containing presets and sub-folders and add it to the selected folder. The importer will ask you whether to destroy the original contents, so that the

new presets replace previous ones, or to keep both.



## Presets list

Presets list contains all presets available in the selected folder. **Double-click** on a preset or use **Load** button to load a preset. Use the buttons at the bottom of the list to perform additional changes. Please note that these are not actual files & folders on disk, but are records in the preset database.

### ♥ Favourite button

Favourite button toggles the 'favourite' indicator for the selected preset.

### ♥ SHOW Show button

Show button shows only the favourite presets and hides the others.

### ↶ SORT Sort button

Sort button shows the presets sorted alphabetically.

### 🎲 RANDOM Random button

Random button selects and loads a random preset from the current folder. This way you can quickly browse the presets in the folder in a completely random order.

### ◀ Previous button

Previous button selects and loads the previous preset from the current folder.

### ▶ Next button

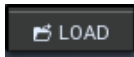
Next button selects and loads the next preset from the current folder.

🎲 SUBMIT AND PERFORM ONLINE PRESET EXCHANGE

## Submit and perform online preset exchange button

Submit and perform online preset exchange button submits the selected preset to the online exchange servers and retrieves all the presets currently in the database. This feature serves as an online database of presets available for all the user community. However it is available

for creators of presets only, meaning that every time you use it, you will need to submit a preset. The preset needs to be yours and must of a reasonable quality; otherwise you may be banned indefinitely from using this service.



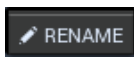
### **Load button**

Load button loads the specified preset. Please note that you can do the same thing by double-clicking the preset itself or pressing the Enter key.



### **Add button**

Add button creates a new preset using the current settings.



### **Rename button**

Rename button lets you rename the selected preset.



### **Replace button**

Replace button replaces the selected preset by one with current settings.



### **Delete button**

Delete button deletes the selected preset.

SEARCH



### **Search**

Search filters the list of available presets to those containing the keywords in name or information.



### **Clear button**

Clear button deletes all text in the search field.



### **Preset**

### **information**

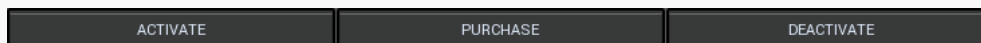
Preset information field contains optional information about the preset, which you can edit when creating or renaming the preset.

# Plugin settings window



Plugin settings window offers more advanced settings and is available via the Settings menu.

## Licence panel



Licence panel lets you manage licences on this computer.

ACTIVATE

### Activate button

Activate button lets you activate your licence for the plugin on this computer.

PURCHASE

### Purchase button

Purchase button navigates to the plugin's website, from which you can purchase a licence for the plugin.

DEACTIVATE

### Deactivate button

Deactivate button lets you deactivate any licences on this computer. It can be useful when you need to work on a public computer or if you sell your licence.

# GUI & Style panel



GUI & Style panel lets you configure the plugin's style (and potentially styles of other plugins) and other GUI properties.

STYLE

## Style button

Style button lets you change the style for this particular plugin.

STYLE WIZARD

## Style wizard button

Style wizard button lets you redefine the default style settings for all plugins.

RANDOM STYLE

## Random style button

Random style button selects a random style with random editor mode.

DEFAULT STYLE

## Default style button

Default style button reverts to the original style and default size of the GUI. Hold **Ctrl** key to revert all MeldaProduction software products, not just the current plugin.

Editor mode

Sliders

Knobs

Buttons

## Editor mode

Editor mode selects the default control used by the plugin editors. Each control is manipulated in a slightly different way, takes a different amount of space and looks different. To make the editor as small as possible, it is usually best to use **Buttons**.



GPU acceleration

Enabled

## GPU acceleration

GPU acceleration controls how much the GPU is used for visual rendering to save CPU power.

**Enabled mode** provides maximum speed and lets the GPU perform as many drawing operations as possible.

**Compatibility mode** uses the GPU for drawing, but doesn't use modern technologies for maximum performance. Use it if you experience occasional problems with drawing, the usual case for older ATI graphics cards. With Pro Tools on OSX this mode is always used instead of Enabled mode due to compatibility problems with this host.

**Disabled mode** disables GPU acceleration completely, drawing is then performed by the CPU. Use only if you experience technical difficulties.

A known problem may occur when using multiple displays with multiple graphical interfaces. When moving the plugin window from one display to another, it may stop displaying correctly until you move it back to the original display.

Frames per second

80

## Frames per second

Frames per second controls the refresh rate of the visual engine. The higher the number is the smoother everything is, but the more CPU it requires. You might want to lower this value if your computer is running out of CPU power.

Enable high DPI support

## Enable high DPI / retina support

Enable high DPI / retina support enables the plugin to use the high resolution on high DPI (Windows) and retina (OSX) devices. It is enabled by default and detected automatically, if the host allows it. If you run into any problems, you can disable it using this option. It may be desired if you use multiple displays where only some of them feature the high resolution making the image on the low resolution ones look ugly.

If you disable this option, on Windows the high DPI device detection will be ignored and the plugin will probably appear very small. You can manually compensate for it by using a bigger style. On OSX disabling this option will disable the high DPI rendering, resulting in the classic blurry look of non-compliant applications. Changes take effect after you restart the host.

Enable colorization

## Enable colorization

Enable colorization enables the plugin to change the colors of certain elements overriding your style settings. Plugins use that to highlight different parts of the graphics interface for easier workflow. You may want to disable it if you just feel it's not for you.

Enable gradients

## Enable gradients

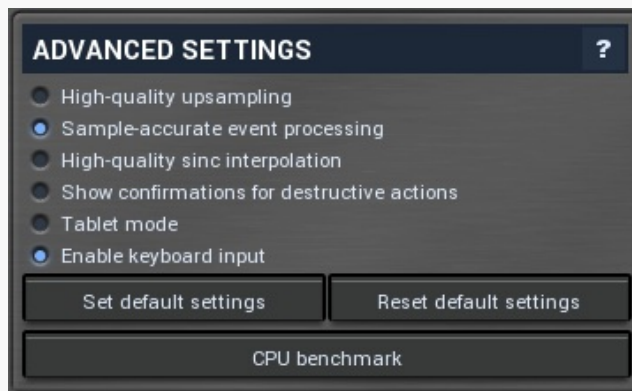
Enable gradients enables allows the plugins to use gradients for various graphs. Disabling this will save some CPU.

Set default editor size

## Set default editor size button

Set default editor size button stores the current editor size as its default. You can drag the bottom-right corner of most plugins to change their size. This can be advantageous as it allows several controls to be bigger and easier to work with. After clicking this item, the current size will be stored and any new instance will open with this size by default. Default sizes are usually the smallest available, so that people with lower resolution displays can still use the plugins. This item is especially useful for users who want to enjoy the advantages of large hi-res displays.

# Advanced settings panel



Advanced settings panel contains settings that control the behaviour of this instance. These are properties that rarely need to be changed, so they have been moved here.

High-quality upsampling

## High-quality upsampling

High-quality upsampling enables the high-quality linear-phase upsampling algorithm. This is relevant only if you use upsampling. Linear-phase upsampling provides the maximum possible quality, however it also requires more CPU and introduces latency. If you need to use upsampling in real-time or want to save resources, you can deactivate this high-quality upsampling option. That will switch to the minimum-phase upsampling algorithm, which offers a superb audio quality as well and does not introduce latency, but it does alter the phase, which may not be acceptable in some cases.

Sample-accurate event processing

## Sample-accurate event processing

### processing

Sample-accurate event processing makes the plugin schedule every event such as MIDI or automation to their accurate locations with sample accuracy, if the host allows it.

For example, if the block size in your host's audio settings is 1024 samples, this means the plugin is probably processing blocks of 1024 samples, in 44100 Hz sampling rate it is about 23ms. If this setting is disabled, any change in automation, MIDI, modulation etc. may then be granularized to 23ms (once per block), which means that you will not be able to recognize events that occur say 10ms apart from each other. When this setting is enabled however, the plugin divides processing blocks to sub-blocks and processes the events at their correct positions. This may, of course, require more CPU power.

High-quality sinc interpolation

## High-quality sinc interpolation

High-quality sinc interpolation activates the maximum audio quality interpolation algorithm, which is used by several modulation processors, such as a flanger or chorus. Unlike standard interpolation sinc interpolation avoids distortion and improves response for high frequencies. The disadvantage is that it introduces some latency and requires much more CPU power.

Show confirmations for destructive actions

## Show confirmations for

## destructive actions

Show confirmations for destructive actions makes the plugin display a confirmation window whenever you are going to change the plugin settings irreversibly when using a feature, for example: when resetting your settings.

Tablet mode

### Tablet mode

Tablet mode enables better support for tablets at the expense of the mouse. Enable this if you are using a tablet to control the plugins and it is behaving incorrectly.

Enable keyboard input

### Enable keyboard input

Enable keyboard input enables the keyboard input for the main plugin window. You may want to disable if the plugin intercepts spacebar key (often used by the host for playback enable/disable and your host doesn't allow for the problem itself).

Set default settings

### Set default settings button

Set default settings button stores the current plugin settings as the defaults, so that when you open a new instance of the plugin, these settings will be loaded automatically.

Reset default settings

### Reset default settings button

Reset default settings button removes the defaults that you set using **Set default settings** button, so that when you open a new instance of the plugin, the factory defaults will be loaded.

CPU benchmark

### CPU benchmark button

CPU benchmark button calculates the performance of the plugin with the current settings.

# Smart interpolation



Smart interpolation panel controls the depth of the smart interpolation algorithm, which controls the parameters in order to provide maximum audio quality and lower the chance of zipper noise. Smart interpolation is engaged whenever you change any parameter via the GUI, modulators, multiparameters, MIDI or automation.

Many parameters can be automated easily and the plugin responds with sample-accurate results. However, several parameters need exhaustive pre-processing when changed. In these cases, the parameters are not updated every sample, but, for example, once every

32 samples. This highly reduces CPU usage, but affects the output quality.

With modulators the situation is more complicated. Besides the updating issue, the modulator itself can perform some pretty advanced processing, hence it is better to perform the processing in blocks. However, the bigger the block, the less often the modulator updates those parameters associated with it and the resulting modulation is less accurate. In a way you can say that the modulator is slower and lazier. This may actually be wanted, so when it comes to modulators it is not true that a better mode always means better output quality.

The smart interpolation mode controls the maximum number of samples being processed before the parameters are updated. **Minimal mode** uses 2048 samples and rarely will do anything unless processing offline. **Normal mode** uses 256 samples and usually is enough to achieve good quality results. **High mode** uses 32 samples and provides perfect quality for most cases. It is also a good compromise between CPU usage and audio quality, so it is the default. **Very high mode** uses 4 samples and you will rarely need it. **Extreme mode** uses 1 sample, which means that everything is updated after every single sample. This provides the highest possible accuracy and quality you can ever achieve, however it requires lots of CPU and it is very unlikely that you will ever need it. If you use this mode and still hear audio artifacts, then either what you are hearing is actually CPU overload, or you are doing something that is not physically possible.

The higher the mode, the quicker the parameter updates, but the more the CPU load.

*Please note that modulating certain parameters without artifacts is impossible. For example, when modulating a delay very quickly, the physics of such a process just cannot occur in the natural world and the results are appropriately unnatural. These physically impossible processes usually manifest themselves as distortion or zipper noise.*

# Modulator



Modulator is an extremely advanced feature, which lets you change parameters automatically depending on various inputs. You can use this to add movement to your sound, respond to some plugins differently for louder sections, or even follow the pitch of the input.

The modulator edit window has two parts: on the left side you can configure the mode of the modulator (the way the modulator works) and on the right side there is a list of parameters to modulate. A modulator can control all automatable parameters (and often more than that) including the parameters of other modulators. Each modulator can control as many parameters as is needed and each of the parameters has its own range and transformation shape. The values and ranges of the first 4 parameters associated with the other modulators can also be modulated/automated. The following modulator modes are available:

**Normal** mode makes the modulator behave like an ordinary low-frequency oscillator (LFO). There are various ways to control its shape as with all oscillators in our plugins. Each modulator can synchronize to the host in the **Synchronization** panel. Modulators can also synchronize with each other using the **Sync groups**. Using **MIDI reset** you can reset the oscillator to any phase using MIDI notes, but obviously to-host synchronization must be disabled in order for this to work.

Note that the settings in this mode are used even if the modulator is actually in a different mode by using "LFO modulation". This basically blends between the actual mode, which may for example detect the input signal level, and give it some additional movement using the LFO depending on the LFO modulation parameter available for each of the remaining modes.

**Follower** mode makes the modulator detect the input signal level. It contains an extremely advanced and accurate level detector taken from our MDynamics plugin. The level follower is an immensely useful feature, yet it may be a little difficult for beginners to comprehend, so we will cover it here in more detail.

It is often necessary to adjust the follower slightly for new material. First, it has the standard



parameters - attack, release, hold and RMS length. These are fairly standard features and help is available for each of them. **Level min and max** controls the range of input levels. When the input level is equal to or below the min level, the modulated parameters' values will be minimal. Similarly, when it reaches the max level, the modulated parameters' values will be at their maximum. This allows for adjustments to the range of input levels, which are certainly different for any audio material and settings. It can be used creatively too - for example, by using very low values for both limits we can differentiate between silent and non-silent parts, similar to the way a gate effect works.

**Advanced detector settings** provide some extraordinary features, such as psycho-acoustic pre-filtering, which forces the modulator to detect loudness instead of raw input levels, custom input signal pre-filtering using a fully featured 6-band equalizer, and custom attack and release shapes. **Band-pass panel** pre-filters the level detection signal using a band-pass filter, so this is like a very simplified version of the equalizer from the advanced detector settings. **Side-chain** makes the modulator measure side-chain input if the plugin has one. For modular plugins the modulator can also be driven by a feedback signal. The **advanced panel** provides some further level processing features that you can take advantage of creatively or to further adjust to your actual audio material.

**Project onto LFO shape** is a more advanced concept, which is available for other modulator modes too. You can easily imagine, that the modulator in any mode generates values for each parameter, we can say it is between 0 and 1, where 0 sets minimum parameter value, and 1 sets the maximum. Project onto LFO shape forces the modulator to use this range in the oscillator shape, which can then be configured in normal mode. The value is basically transformed by the oscillator shape, where the values generated by the modulator are on the horizontal axis (phase) and the output is the actual oscillator value. This feature has no physical meaning and can only be used creatively - to transform the more or less linear results of the level follower into a much more complicated curve.

*Let us demonstrate the follower mode with an example - the idea is to apply a delay to a snare drum within a previously mixed drumset. This is commonly used on reggae/dub rhythms for example, however in these cases the snare track is usually available separately. Using the modulators you can get somewhat interesting results even with an already mixed drumset. The idea is to increase the input gain whenever the snare is playing, so that only the snare drum (and potentially other instruments playing at the same moment) are passed into the delay. So first teach the modulator to control input gain parameter of the delay and set it to follower mode, potentially configure some of the parameters to get the desired response. Now the louder the input is, the more delay you get. To make it respond only to snare drum, enable the band-pass and set the filter limits accordingly, e.g. 500Hz to 1k. This makes the input gain increased depending on the input level in this part of the spectrum, which contains the snare drum.*

**Envelope** mode causes the modulator to generate an arbitrary envelope, similar to those from synthesizers. It can either follow MIDI - the envelope starts when a key is pressed, goes through the attack and decay stages, then holds in sustain stage until the key is released when the release stage begins, or it can follow audio - when the audio level exceeds **Threshold on** it behaves the same way as when a note is pressed in MIDI mode, and then when the input level drops below **Threshold off** it behaves like a key release. As with most modes there is LFO modulation and LFO projection and the input level can be driven by the side-chain or feedback if available. The envelope shape can be adjusted using several controls (lengths of each stage etc.) and you can even draw your own shape.

**Random** mode is a smooth random generator. It is very handy if you want some parameters to change over time, but do not actually want them to be periodic like LFOs. A modulator in random mode does not actually generate random values, the results will always be the same at each position in your arrangement in the host. This allows a pseudo synchronization with the host and ensures a "what you hear is what you get" performance. **Speed** parameter controls the speed of change and any slight change to this parameter will change the whole stream.

**Pitch** detects the pitch of the input signal assuming it is not polyphonic (here it can work too and will probably detect the lowest note, however it is definitely not suitable for percussive signals, which do not have a pitch). It is very useful, enabling you to tune an oscillator to follow your singing, or allow an equalizer to control separate harmonics of a vocal, use a distortion to get more drive for higher notes in a guitar solo and much more. The pitch detection may be a little tricky to understand, so we will discuss it in more detail.

A pitch detector takes the input signal and tries to approximate the pitch of the fundamental frequency in it. It is physically impossible to detect pitch instantly, as an extreme example, 20Hz takes 50ms for the signal to evolve enough to detect that there is actually a 20Hz frequency in the signal. For this and many other reasons any pitch detector employs several limitations. These are available in the **Detector panel**. The defaults will work well for most audio material, however, it is useful to understand the parameters, so that you can let the detector adapt better to your particular audio materials if necessary, and also in order to be more creative.

**Min and max frequency** parameters in the Detector panel control the limits of the frequencies you expect in the input. For example, a female voice is unlikely to sing below 100Hz, so it is customary to set the minimum frequency to 100Hz or even higher. Voice signals contain several artifacts, blows and pops, all of which can temporarily create frequencies below the actual pitch of the voice, so setting these limits is preferable to avoid "jumps" to incorrect pitches. **Stabilization** and **Speed** also prevent these jumps by restricting how quickly the pitch can change. These can also be used creatively. **Threshold** controls the minimum level of the input signal to be considered "not-silent and probably having pitch". This acts as a form of gate, which prevents the detector from analyzing irrelevant rumble in between actual performances. **Shift panel** allows the detected pitch to be shifted up or down and **Auto-tune panel** moves it to the closest note - similar to the automatic pitch changing function from MAutoPitch, except no pitch shifting is actually done and the results are used purely to control some parameters.

**Min and max frequency** parameters in the top of the editor have a very different meaning than the parameters of the same name in the detector panel. From now on we will assume that the pitch has been detected successfully and are now considering what to do with the results. Again, we may assume the modulator generates values from 0 to 1, where at 0 the modulated parameters' values become minimal and reach maximum at 1. When the input pitch is equal or below the min frequency parameter, the modulator's value is 0, hence modulated parameters will have a minimal value as well. Similarly when the pitch reaches max frequency, the modulated parameters will get to the maximum.

Now you may say this makes no sense, because the detected pitch cannot exceed the limits specified in the **Detector panel** anyway. The reason for this is that most "frequency" parameters of all plugins are limited from 20Hz to 20kHz, whether it is the frequency of a band in an equalizer, or a high-pass frequency in a phaser for example. It is a reasonable solution since physiologically speaking these figures are on or around the range of our hearing limits.



*Let us explain the concept with an example. We want to modulate a band of an equalizer, so that it always follows the fundamental frequency, the pitch, of our audio material. All we need to do is to switch the modulator to pitch mode, allow it to control the band frequency parameter and set the range for this parameter to the full range, from 20Hz to 20kHz. The pitch detector may then detect frequencies from 50Hz to 2kHz, but the modulator takes it that the actual limits (converted to 0..1) are 20Hz to 20kHz and that exactly the same range is configured for the band frequency parameter, so you could say that "they understand each other". We did not need to touch the min and max frequency parameters at all.*

*Here is one more example, where we would actually want to adjust the min and max frequency parameters. We want to control a drive parameter of a distortion for a guitar so that the higher the guitarist plays the more distortion he gets. Again, we teach a modulator to control the drive parameter, for any range we want, and switch the modulator to pitch mode. Now the modulator will move the drive parameter, but only slightly, because it assumes the pitch can vary from 20Hz to 20kHz, but the guitar may actually only play from about 100Hz to 1kHz. So we can use the min and max frequency parameters to say "what is high and what is low", to limit the frequency range. There are no general rules here, you have to experiment, because every instrument and parameter is different.*

To sum things up, the difference between controlling a frequency parameter and a drive parameter is simply the fact that a frequency parameter is compatible with the pitch. After all, pitch is nothing more than a frequency (strictly speaking it is a logarithmic representation of frequency).



## Random button

Random button generates random settings. Note that unlike copy & paste, presets & randomization do NOT affect the set of parameters being modified, hence it serves to optimize adjustment of the modulator behaviour assuming that you already specified the set of parameters to control.

If you hold **Shift**, the plugin will undo previous randomization.



## Presets button

Presets button displays a window where you can load and manage available presets. Hold **Ctrl** when clicking to load a random preset instead.



## Left arrow button

Left arrow button loads the previous preset.



## Right arrow button

Right arrow button loads the next preset.



## Randomize button

Randomize button loads a random preset.



## R button

R button enables automation read. This way you can actually automate the modulation value. First you use **W button** to record the modulator values over time. After that you can modify it in some way and enable automation read to override the normal modulator behaviour. Note that the results may be different when automation is used with potentially lower audio quality

and slower response.

## **W** **W button**

W button enables automation write. This way you can actually automate the modulation value. Use the button to record the modulator values over time. After that you can modify it in some way and enable automation read to override the normal modulator behaviour. Note that the results may be different when automation is used with potentially lower audio quality and slower response.

# Parameters panel



Parameters panel contains the list of the parameters that the modulator is controlling, their ranges etc.

## **⋮** **Presets button**

Presets button displays a window where you can load and manage available presets. Hold **Ctrl** when clicking to load a random preset instead.

## **◀** **Left arrow button**

Left arrow button loads the previous preset.

## **Right arrow button**

Right arrow button loads the next preset.



## **Restore original values when disabled**

Restore original values when disabled makes the modulator restore the original parameter values when it is disabled by automation or modulation. Normally when you manually disable the modulator, the original values are restored as that is usually desired. However when you control the modulator enable state by automation or modulation, you may or may not want this to happen.

## **button**

This button adds a parameter to the list of controlled parameters. Alternatively you can use the learn feature available by right-clicking the modulator button.

## **button**

This button deletes the selected parameter from the list of controlled parameters.

## **Parameter**

Parameter defines the target parameter which is being modulated. The set contains all automatable parameters.

## **Range mode**

Range mode defines how the parameter range is selected. While sometimes it is better to specify minimum and maximum, other times it is better to use a nominal center and depth (% of full scale). This control allows you to define which one it will be.

**Up and down** mode makes the values go above and below the selected **Value**, which is considered the center. The interval is made smaller if necessary.

**Full range mode** is similar, except the range is symmetrically constrained, so the selected **Value** may not be the center anymore.

**Up/down only modes** goes from the selected value up/down only.

Let's compare these 4 modes. Taking a value of -12dB value, with a depth of 75% and a scale of +/- 24dB. The nominal range is therefore = +/-24 dB \* 75% = 36dB. With values of 0%, 50% and 100% the outputs are:

Up and down: -24, -12, 0 (range constrained to 12 dB either side)

Full range: -24, -6, 12 (range limited to minimum, but not constrained)

Up only: -12, 6, 24 (range not constrained = +/-24 dB \* 75% = 36dB)

Down only: -12, -18, -24 (range limited to minimum)

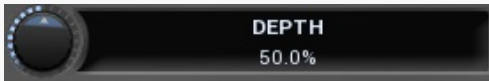
**Interval mode** is the most simple one and goes from **Value** to **Maximal value**.

## **Value**

Value defines the center of the target parameter's range or the minimum if the **Range mode** is set to **Interval**.

## **Maximal value**

Maximal value defines the upper limit of the target parameter's range. It is available only if the **Range mode** is set to **Interval**. This value can be lower than **Value**. 0% is always mapped to reference>Value and 100% to reference>Maximal value.



## Depth

Depth defines size of the target parameter's range. It is used only if the **Range mode** is not set to **Interval**.



## Invert

Invert checkbox inverts the target parameter's range, so that minimum becomes maximum and vice versa.



## Use first parameter's range

Use first parameter's range makes the parameter display use the same range as the first parameter in the list. This is often useful if want to control the range in some way and apply the range to multiple parameters.



## Show transformation shape button

Show transformation shape button displays the graph editor, which lets you tweak the shape of the curve used to control the selected parameter. The X axis shows the original values, the Y axis defines the results. Note that this takes some CPU, therefore you have to enable it using the enable button in the title.

NORMAL

ENVELOPE

RANDOM

## Mode

Mode defines the way in which the modulator works. The modulator is like a black box that generates one number in range 0% to 100% at each moment and then assigns the appropriate value to each of the target parameters. The mode defines what this number will be. Select the particular tab to control the modulator's behaviour.

**Normal** mode uses a standard low-frequency oscillator (LFO) to drive the parameters.

**Follower** mode uses the level of the input signal.

**Envelope** generates an envelope using MIDI notes or by following input signal level.

**Random** generates randomized output which is however the same every time you render the song.

**Pitch** detects and follows the pitch of the input signal.

# Normal mode



Normal mode makes the modulator work as a traditional low-frequency oscillator (LFO). Note that even if the modulator itself is running in a different mode, you can still blend this LFO using the **LFO modulation** parameter available on each tabbed page. The LFO parameters themselves are available on the first tabbed page only though.

## Signal generator



Signal generator defines the modulation LFO shape. It is used by the LFO generator, but also for the **Project** feature. Signal-generator is an incredibly versatile generator of low & high frequency signals. It offers 2 distinct modes - Normal and Harmonics. **Normal mode** is appropriate for low-frequency oscillators, where the graphical shape is relevant and is used to drive some form of modulation. For example, a tremolo uses this modulation to change the actual signal level in time. Frequencies for such

oscillators usually do not exceed 20Hz as this is a sort of limit above which the frequencies become audible.

**Harmonics** mode is designed for high-frequency oscillators, where the actual shape is not as important as the harmonic content of the resulting signal, hence it is especially useful for actual audio signals. Please note that since a shape can contain more harmonics than those available from the harmonic generator, the results may not be exactly the same. As an example, a rectangular wave in normal mode may sound fuller than when converted to the harmonic mode.

Use the arrow-down button to switch from normal mode to harmonics mode or click the **Normal** and **Harmonics** buttons

## Normal mode

The generator first uses a set of predefined signal shapes (sine, triangle, rectangle...), which you can select directly by right-clicking on the editor and choosing the requested shape from the menu. This menu also provides a link to the modulator shapes preset manager, normalization and randomization. You can also use the **Main shape** parameter, which generates a combination of adjacent signals to provide a nearly inexhaustible number of basic shapes.

The engine then combines the predefined shape with a **Custom shape**, which may be anything you can draw using the advanced envelope engine, depending on the level set by the **Custom shape** control. Use the **Edit** button to edit the custom shape.

You can also combine those results with a fully featured step sequencer, with variable number of steps and several shapes for each of them, depending on the level set by the **Step sequencer** control. Use the lower **Edit** button to edit the step sequence.

Those results may be mixed with a custom sample, which is available from the advanced settings, accessed by clicking the **Advanced** button.

**Smoothness** softens any abrupt edges, generated by the step sequencer for example.

Finally there are **Advanced** features providing more complex transformations, adding harmonics etc. or you can click the **Randomize** button in the top-left corner to generate a random, but reasonable, modulator shape.

## Harmonics mode

Harmonics mode represents the signal as a series of harmonics (that is, multiples of the base frequency). For example, when your oscillator has a frequency of 2Hz (set in the panel), then the harmonics are 2Hz, 4Hz, 6Hz, 8Hz etc. In theory, any signal can be created by mixing a potentially infinite number of these harmonics.

The harmonics mode lets you control the levels and phases of each harmonic. The top graph controls the levels of individual harmonics, while the bottom one controls their



phases. Use the left-mouse button to change the values in each graph, the right-mouse button sets the default for the harmonics - 0% level and 0% phase. In both graphs the harmonics of power 2 (that is octaves) are highlighted. Other harmonics may actually sound disharmonic, despite their names.

*For example, if you reset all harmonics to the defaults and increase only the first one, you will get a simple sine wave. By adding further harmonics you make the output signal more complex.*

**Harmonics** controls the number of generated harmonics. The higher the number is, the richer the output signal is (unless the levels are 0% of course). This is useful to make the sound cleaner. For example, if you transform a saw-tooth wave to harmonics, it would not sound like a typical saw-tooth wave anymore, but more like a low-passed version of one. The more harmonics you use, the closer you get to the original saw-tooth wave.

**Generator** is a powerful tool for generating the harmonics, which are otherwise rather clumsy to edit. The generator provides several parameters based upon which it creates the entire series of harmonic levels and phases. These parameters are usually easier to understand than the harmonics themselves. Part of the generator is the randomizer available via the **Random seed** button, which smartly generates random settings for the generator. This makes the process of getting new sounds as simple as possible.

## Signal generation fundamentals

The signal generator produces a periodic signal with specified wave shape. This means that the signal is repeating over and over again. As a result it can only contain multiples of the fundamental frequency. For example, if the generator is producing 100Hz signal, then it can contain 100Hz (fundamental or 1st harmonic), 200Hz (2nd harmonic), 300Hz (3rd harmonic), 400Hz (4th harmonic) etc. However, it can never produce 110Hz. You can then control the level of each harmonic and their relative phases. It does not matter whether you use the normal mode using oscillator shapes, or harmonics mode where you can control the harmonics directly. If both modes result in the same wave shape (such as sine wave vs. 1st harmonic only), then the result is exactly the same.

Sine wave is the simplest of all as it contains the fundamental frequency only. The "sharper" the signal shape is, the more harmonics it contains. The biggest source of higher harmonics is a "discontinuity", which you can see in both rectangle and saw waves. In theory, these signals have an infinite number of harmonics. However since our hearing is highly limited to less than 20kHz, the number of harmonics which are relevant is actually pretty small. If you generate a 50Hz signal, which is very low, and assuming that you have extremely good ears and you actually hear 20kHz, then the number of harmonics audible for you is  $20000 / 50 = 400$ .

### What happens above 20kHz?

Consider the example above again, what happens with harmonics above 400? These either stay there and simply are not audible, disappear if anti-aliasing is used, or get



aliased back under 20kHz in which case you get the typical digital dirt.

When you convert a rectangle wave to harmonics mode, only the first 256 harmonics are used, so it basically works like an infinitely steep low-pass filter. What is the limit then?  $50 \text{ Hz} * 256 = 12.8\text{kHz}$ . The harmonic mode will not produce anything above this limit if you are generating a 50Hz signal. Most people do not hear anything above 15kHz, so this is usually enough, but if not, you may need to use the normal mode where you get the "infinite" number of harmonics.

## **What you see is not always what you get!**

Say you want a rectangle wave and play a 440Hz tone(A4). You would expect the output signal to be a really quick rectangle wave, right? Wrong! If you would do that, and actually most synthesizers on the market do that, you would get the infinite number of harmonics. And, since you are working in say 48kHz sampling rate, the maximum frequency that can actually exist in your signal is 24kHz. So everything above it would get aliased below 24kHz, and there would be a lot of aliased dirt.

The "good" synthesizers perform a so-called anti-aliasing. There are several methods, most of them require quite a lot of CPU or have other limitations. The goal is to remove all frequencies above the 24kHz in our case or in reality, it is more about removing all aliased frequencies above 20kHz - this means, that we do not care about frequencies above 20kHz, because we do not hear them anyway. But we will keep it simple. Let's say we remove everything above 20kHz. You already know that the rectangle wave can be created using an infinite number of harmonics or sine waves. We removed everything above the 45th harmonic ( $20000 / 440$ ) so our rectangle wave is trying to be formed using just 45 harmonics, so it will not really look like a rectangle wave.

After some additional filtering (like DC removal), the rectangle wave may look completely different than a true rectangle wave, yet it would sound the same! Does it matter? Not really. You simply edit the shape as a rectangle wave and let the synthesizer do the ugly stuff for you. But do not check the output, because it may be very different than what you would expect ;).

## **How can I generate non-harmonic frequencies?**

Ok, so now you are playing a 440Hz (A4) saw wave, it contains 440Hz, 880Hz, 1320Hz etc. Anything generated using the signal generator can contain only these frequencies, the only difference is the levels and phases of each of them. What if you want to make the signal dirty by adding say 500Hz? Well, that is not that simple! Here we are getting into audio synthesizer stuff, so let us just give you a few hints.

The traditional way is to use modulation. One particular method is called frequency modulation (FM). Instead of generating a 440Hz saw wave with your generator, you change the pitch, up and down. You are modulating the frequency, that's why FM. It is basically a vibrato, but as you increase the speed of the vibrato, it gets so quick that you stop noticing the pitch changes (that's very simplified but it serves the purpose) and instead it starts producing a very complex spectrum. Will the 500Hz be there? Well, if setup correctly, yes, but there will also be lots of other non-harmonic frequencies.

Another way is possible without any other tools. Let's say you do not want 440Hz, but 660Hz. Then you may generate 220Hz instead of 440Hz (which is one octave below it) and voila, 660Hz is the 3rd harmonic ( $3 \times 220$  is 660)! But you need to shift the saw wave one octave above. Fortunately it is not that hard here - go to the normal mode, select saw tooth, click advanced, and use the harmonics panel to remove the fundamental and leave just the 2nd harmonic, then convert it to harmonic mode. Well, it's not that hard, but it's not exactly simple either...

The only way is, of course, additive synthesis. In that case you do not use one oscillator, but many of them. It lets you generate just about anything. But there is a catch, actually many of them. First, you need to say "ok I want this frequency and that frequency...", the setup is actually infinitely hard as there may be an infinite number of frequencies :). And the second is, of course, CPU requirements.

So is there some ultimate solution? Nope, sorry. The good thing is, you will not probably need it, because while what you see is not always what you get, also what you want is often not what you really want to hear :).

### **Random button**

Random button generates random settings using the existing presets.

### **Presets button**

Presets button displays a window where you can load and manage available presets. Hold **Ctrl** when clicking to load a random preset instead.

### **Left arrow button**

Left arrow button loads the previous preset.

### **Right arrow button**

Right arrow button loads the next preset.

### **Randomize button**

Randomize button loads a random preset.

### **button**

This button copies the settings onto the system clipboard.

### **button**

This button loads the settings from the system clipboard.

### **Randomize button**

Randomize button generates random settings.

NORMAL

### **Normal button**

Normal button switches the generator into the normal mode, which lets you edit the shape of the oscillator. This is especially advantageous for low-frequency oscillators, where the shape matters even though it doesn't have any physical meaning.

## ↓ Convert button

Convert button converts the current shape into harmonic-based representation. Please note that since the number of harmonics is limited, the result will not perfectly resemble the original shape.

## HARMONICS

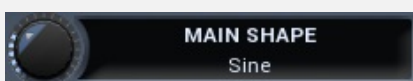
## Harmonics button

Harmonics button switches the generator into the harmonics mode, which lets you edit the levels and phases of individual harmonics. This is especially advantageous for high-frequency oscillators, hence sound generators.

## Signal generator in Normal mode

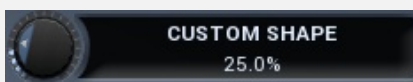


Signal generator in Normal mode works by generating the oscillator shape using a combination of several curves - a predefined set of standard curves, custom shape, step sequencer and custom sample. It also post-processes the shape using several filters including smoothing to custom transformations. This is especially useful when using the oscillator as an LFO (low-frequency-oscillator), where the harmonic contents does not really matter, but the shape does.



## Main shape

Main shape controls the main shape used by the signal generator. There are several predefined shapes, such as sine, triangle or rectangle, which you can choose from or even interpolate between using this control.



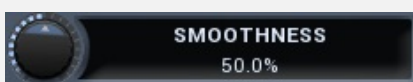
## Custom shape

Custom shape controls the amount of the custom shape that is blended into the main shape.



## Edit button

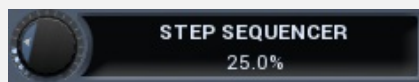
Edit button shows the custom shape editor.



## Smoothness

Smoothness controls the amount of smoothing. Many shapes, especially those produced by the step sequencer, have rough jagged edges, which may be

advantageous, but when used to modulate certain parameters, the output may be clicking or causing other artifacts. Smoothness helps it by smoothing the whole signal shape out and removing these rough edges.



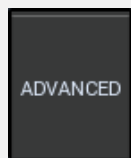
## Step sequencer

Step sequencer controls the amount of the step sequencer shape that is blended into the main shape (which has already been blended with the custom shape).



## Edit button

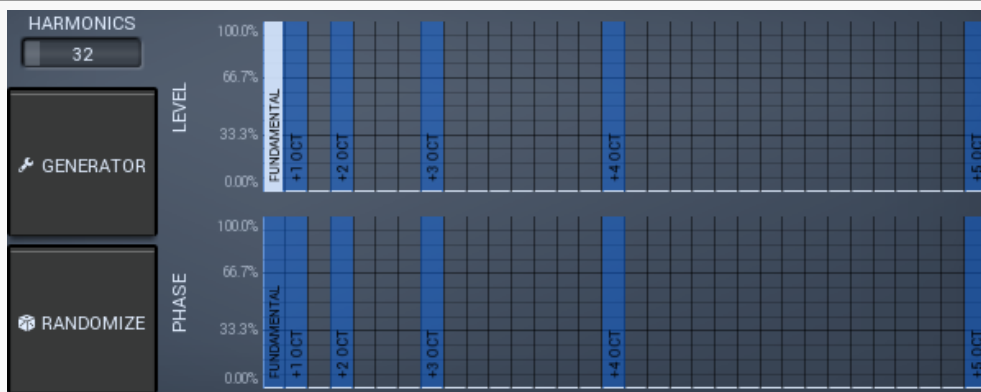
Edit button shows the step sequencer editor.



## Advanced button

Advanced button displays an additional window with more advanced settings for post-processing the signal shape, such as harmonics or custom transformations.

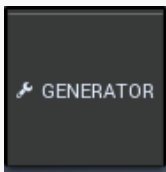
# Signal generator in Harmonics mode



Signal generator in Harmonics mode works by generating the oscillator shape using individual harmonics. Essentially a harmonic is a sine wave. The first harmonic, known as the fundamental, fits once in the oscillator time period, hence it is the same as selecting sine wave in the **Normal mode**. The second harmonic fits twice, the third three times etc. In theory, any shape you create in normal mode can be converted into harmonics. However, this approach to signal generation needs an enormous number of harmonics, which is both inefficient to calculate and mostly hard to edit. Therefore, the harmonic mode can process up to 256 harmonics, which is enough for very complex spectrums, however it is still not enough to generate an accurate square wave for example. If your goal is to create basic shapes, it is better to use the normal mode.

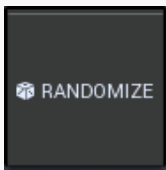
It is nearly impossible to say how a particular curve will sound when used as a high-frequency oscillator in a synthesizer, just by looking at its shape. Harmonics mode, on the other hand, is directly related to human hearing and makes this process very simple. In general, the more harmonics you add, the richer the sound will be. The higher the harmonic, the higher the tone. Usually, one leaves the first harmonic enabled too, as this is the fundamental tone, however you may experiment with more dissonant sounds without it.

Editing harmonics can be time consuming unless you hear what you want, so a signal generator is also available. This great tool lets you generate a random spectrum by a single click. You can also open the **Generator<\reference> settings and edit its parameters, which basically control the audio properties in a more natural way - using parameters such as complexity, harmonicity etc.**



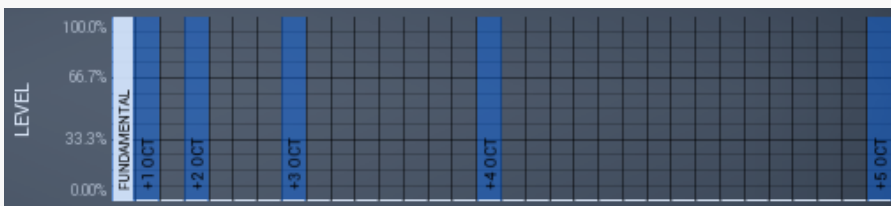
## button

This button shows a powerful harmonics generator, which can create unlimited number of various timbres and even analyze a sample and extract harmonics from it.



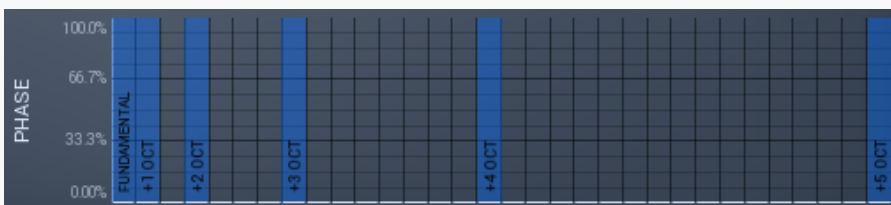
## Randomize button

Randomize button selects random parameters for the harmonics generator, so you can use it to get a random sound character instantly. Hold **Ctrl** to slightly modify existing generator settings instead of completely changing them.



## Magnitudes graph

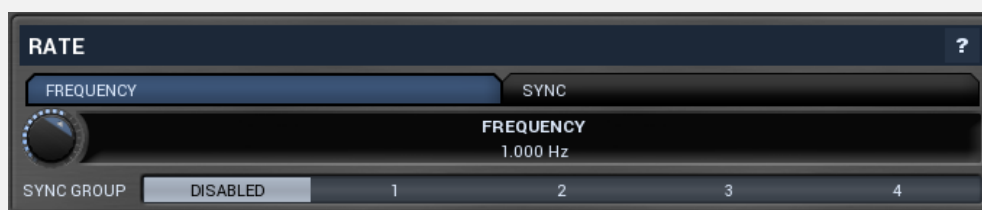
Magnitudes graph contains the levels of the individual harmonics. The highlighted bars are octaves, thus the 1st, 2nd, 4th, 8th harmonic etc.



## Phases graph

Phases graph contains the phases of the individual harmonics. The highlighted bars are octaves, thus the 1st, 2nd, 4th, 8th harmonic etc.

## Rate



FREQUENCY

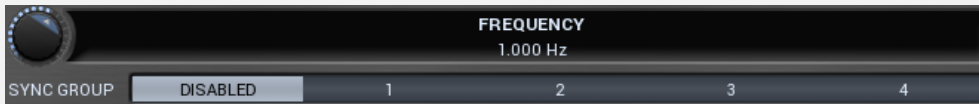
SYNC

## Rate panel

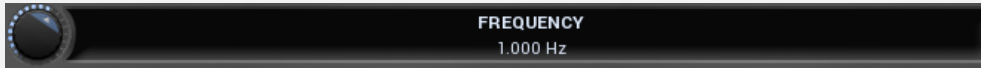
Rate panel contains parameters controlling the speed of the LFO, whether the modulator is set to **Normal** mode or any other mode while the **LFO modulation** is

used.

## Frequency panel

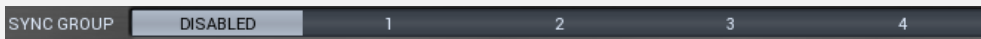


Frequency panel controls the LFO frequency and is available only when the to-host synchronization is off.



### Frequency

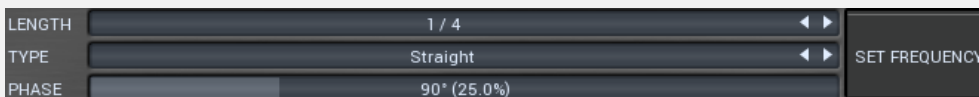
Frequency defines the modulation speed.



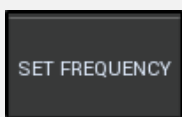
### Sync group

Sync group lets you synchronize the modulators with each other and potentially with other parts of the plugin. It can be controlled only when **to-host synchronization** is disabled, otherwise it is overridden by synchronization from the host. By using the same synchronization group for all modulators you ensure they will always be in-sync even though no other synchronization is used. This can be useful, for example, when you want to modulate different parameters with different shapes or when using some more advanced method, such as using a follower. When the synchronization is enabled, it works on the 'first is the leader' basis, hence the first modulator controls the rest of the modulators in the same group.

## Synchronization panel



Synchronization panel contains parameters for the to-host synchronization.



### Set frequency button

Set frequency button sets the **Frequency** parameter available for the frequency mode so that it matches the current synchronization. That way you can set the modulator's frequency to the current synchronization and then change it a little for example.

## MIDI reset panel



MIDI reset panel configures the MIDI reset feature, which will reset the oscillator when a MIDI note is received. This way you can make the oscillator perform "in-sync" with your playing. Note that once you enable it, the oscillator will not be in phase-sync with the host.

### **ENABLE** button

This button enables or disables the feature.

### **SINGLE SHOT** Single shot

Single shot button activates the single shot mode in which the LFO doesn't cycle around but instead only goes once from left to right, then stops until the MIDI reset occurs.

### **SINGLE SHOT RESET** Single shot reset

Single shot reset button defines if the phase should reset to 0 after a single shot period ends. For most waves such as sine it doesn't really matter since the value at 0 (the start of the cycle) is the same as value at 1 (the end of the cycle). But it might matter for saw wave for example.

### **NOTE ON** Note-on

Note-on controls if the MIDI reset should occur when a note is pressed.

### **NOTE ON ONLY FIRST** Note-on only first

Note-on only first controls if the MIDI reset should occur when a note is pressed only if it is the first note (thus no other note is being held).

### **NOTE OFF** Note-off

Note-off controls if the MIDI reset should occur when a note is released.

### **NOTE OFF ONLY LAST** Note-off only last

Note-off only last controls if the MIDI reset should occur when a note is released only if it is the last note (that is, no other note is being held afterwards).

### **MIN VELOCITY** Min velocity

Min velocity defines the minimum velocity that will reset the oscillator.

### **MAX VELOCITY** Max velocity

Max velocity defines the maximum velocity that will reset the oscillator.

### **MIN NOTE** Min note

Min note defines the minimum note that will reset the oscillator.



MAX NOTE  
127

## Max note

Max note defines the maximum note that will reset the oscillator.

PHASE  
0° (0%)

## Phase

Phase defines the initial oscillator phase after a reset.

CHANNEL

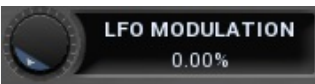
## Channel

Channel defines note MIDI channel to reset the oscillator.

# Envelope mode

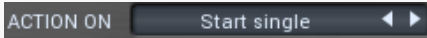


Envelope mode makes the modulator generate arbitrary envelopes from input MIDI or by analyzing the audio input level. When using MIDI the modulator responds to input note-on and note-off messages. When using an audio input (if available), the modulator detects the input level and when it exceeds the **Threshold On**, it behaves like a note-on MIDI event. Afterwards when the level drops down below **Threshold Off**, it behaves like a note-off MIDI event. Each event can result in just about any action. By default note-on starts the envelope and note-off initiates the release stage, but a different behaviour is also possible depending on **Action ON** and **Action OFF** parameters.



## LFO modulation

LFO modulation defines the amount of LFO modulation applied in addition to the envelope. With 0% the modulator uses only the envelope; with 100% the modulator does the same job as if the modulator were in **Normal** mode. To set the LFO parameters switch to normal mode temporarily.



### Action ON

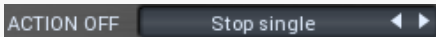
Action ON controls what happens when a note-on event occurs (either via audio or MIDI). **Start single** action, which is the default, means that the envelope will start on the note-on event, but only once, it won't start again until you release all of the keys that are relevant for MIDI triggering only.

**Start** makes the envelope start every time you press a key, whether another key is already pressed or not. The envelope will seamlessly jump to the attack stage avoiding any abrupt changes.

**Start forced** is similar, but lets the envelope start from the very beginning every time. So for example if the envelope is currently in a long release stage and the new modulator value is 0.5, then the Start action jumps to a location in the attack stage where there is a value of 0.5 as well, hence avoiding abrupt changes. Start forced action on the other hand starts the whole envelope over from the beginning of the attack stage, where the value is most likely 0.

**Ignore** action simply ignores this event.

The remaining actions are rather creative and let you do the opposite - initiate release stage and stop the envelope when you press a key.



### Action OFF

Action OFF controls what happens when a note-off event occurs (either via audio or MIDI).

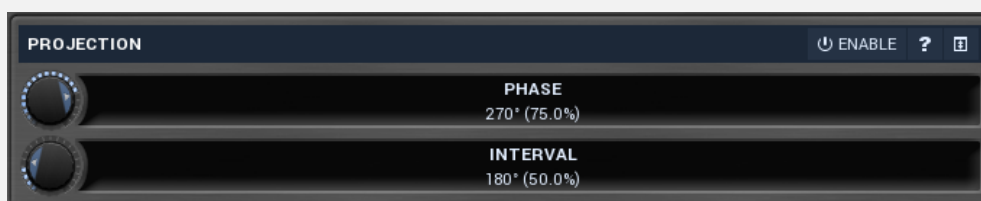
**Stop single** action, which is default, means that the envelope will enter the release stage on the note-off event, but only once, at the moment you release the last key (if you were holding more than one) it is relevant for MIDI triggering only.

**Stop** makes the envelope enter the release stage every time you release a key, whether another key is already pressed or not.

**Ignore** action simply ignores this event.

The remaining actions are rather creative and let you do the opposite - start the envelope when you release a key.

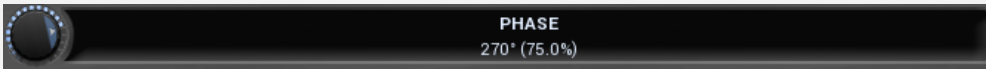
## Projection panel



Projection panel contains parameters of projection onto the LFO oscillator shape, which takes the value generated by the modulator and puts it onto the LFO oscillator shape. This features is useful for several creative effects.

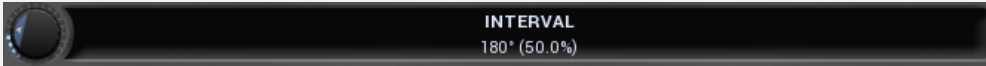
## **button**

This button enables or disables the projection onto the LFO oscillator shape.



## **Phase**

Phase defines the offset from zero of the signal curve. By default it is 75%, because when you look at common oscillator shapes, such as a sine or triangle, at position 75% its value is minimal. Then when you look at the right side, the value is growing up to the 25%, where it becomes the maximum.



## **Interval**

Interval defines the size of the interval from the oscillator shape in addition to **Phase**. As a result, phase defines where you start on the shape and interval specifies size of the window on the shape. Default value is 50% as for example sine grows from minimum to maximum in 50% of the period.

# ADSR



## **Random button**

Random button generates random settings using the existing presets.

## **Presets button**

Presets button displays a window where you can load and manage available presets. Hold **Ctrl** when clicking to load a random preset instead.

## **Left arrow button**

Left arrow button loads the previous preset.

## **Right arrow button**

Right arrow button loads the next preset.

## **button**

This button copies the settings onto the system clipboard.

## **button**

This button loads the settings from the system clipboard.

## Custom shape **Custom shape button**

Custom shape button enables custom shape mode, which lets you draw your own attack and release stages using the envelope system. Both stages are then automatically connected to form the resulting envelope.

## Immediate release **Immediate release button**

Immediate release button activates the immediate release mode in which case the note-off causes an immediate switch to the release stage. If this is disabled, the release stage does not occur until the whole attack/decay stage finishes.

## **Sync button**

Sync button controls the ADSR tempo sync feature. By default this is disabled and means that all times are followed exactly, meaning that if **Attack** is say 100ms, then it will be 100ms indeed. Tempo sync lets the plugin adjust the times to ensure it will be always in sync with the host tempo. In this case 100ms may become say 125ms if the tempo is 120bpm, because 125ms is the length of a 16th note. This makes it extremely simple to convert any envelope to a tempo-synced one. The plugin always chooses the nearest longer note, in other words it always round up.

**Straight** and **Triplets** modes automatically find 'nice' values.

*For example, if a 16th note takes 100ms, the attack time is 550ms, and the sync mode is straight, then the plugin checks for 100ms, find out that it is too low, so it checks 8th note, being 200ms, still too low, then continues with quarter note, which takes 400ms, and still not enough, finally 800ms corresponding to a half note is the one, so the resulting time will be 800ms. Triplet cases are more complex, but the principle is the same.*

**1/16**, **1/8** and **1/4** modes choose the nearest higher multiply of the base note length. For example, if a 16th note takes 100ms, the attack time is 550ms, and the sync mode is 1/16, the resulting time will be 600ms.

## **Settings button**

Settings button displays additional tremolo settings, containing tremolo behaviour and shape.

# Random mode



Random mode makes the modulator generate a pseudorandom sequence. Please note that despite its name, it is created so that it generates the same sequence every time. However the generator is linked to the **Speed** parameter, so if you change it, the whole sequence changes.

## Mode

Mode defines the behaviour of the randomizer.

**Smooth** produces a continuous random modulation. **Smoothness** then controls how smooth it will be, where 0% means it will connect distinct values by straight lines, 100% means the modulation will be a completely smooth curve walking through these random points.

**Steps** produces a step change every particular time interval. It can also granularize it to a specified number of possible values according to **Smoothness** value. 100% disables the granularization. Otherwise the number of steps is the number of percentage values, so 3% means there will be 3 possible values, equally distributed over the range, let's call them 0%, 50% and 100%. Since it doesn't make sense to have 0 or 1 steps, the minimum is always 2. 2 steps essentially means the modulator is randomly switching between the minimum and maximum values for all associated parameters.

**Change on MIDI note** generates a random value every time a MIDI note is received by the plugin.

## LFO modulation

LFO modulation defines the amount of LFO modulation applied in addition to the random generator. With 0% the modulator uses only the randomizer; with 100% the modulator does the same job as if the modulator were in **Normal** mode. To set the LFO parameters switch to normal mode temporarily.

## Speed

Speed defines the speed of the random changes proportional to the current tempo. 0% means that the speed is the same as your song's tempo.

## Smoothness

Smoothness defines the amount of smoothing of the randomizer curve in order to minimize abrupt edges.

## Synchronize to LFO

Synchronize to LFO lets you synchronize the speed of the random sequence to LFO (Normal mode), hence also to your host. **Speed** is still applicable and, for example, +100% means 2x speed, +200% means 4x the speed etc.

## True random

True random makes the modulator produce a true pseudo-random sequence independent of the current position within the project. By default this is disabled, so that every time you play your project, it sounds the same. But you might want to enable this option, for live performances for example.

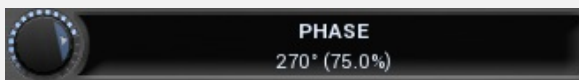
## Projection panel



Projection panel contains parameters of projection onto the LFO oscillator shape, which takes the value generated by the modulator and puts it onto the LFO oscillator shape. This feature is useful for several creative effects.

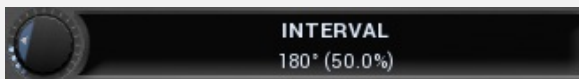
### **ENABLE** button

This button enables or disables the projection onto the LFO oscillator shape.



### **Phase**

Phase defines the offset from zero of the signal curve. By default it is 75%, because when you look at common oscillator shapes, such as a sine or triangle, at position 75% its value is minimal. Then when you look at the right side, the value is growing up to the 25%, where it becomes the maximum.



### **Interval**

Interval defines the size of the interval from the oscillator shape in addition to **Phase**. As a result, phase defines where you start on the shape and interval specifies size of the window on the shape. Default value is 50% as for example sine grows from minimum to maximum in 50% of the period.



# Arpeggiator



Arpeggiator is a powerful sequencer which transforms the notes that you are playing into potentially long evolving sequences. The idea came from guitar arpeggios, where the left hand is holding the strings, defining the notes to be played, but the right hand is actually playing them, in any order.

This arpeggiator is based on the sequencer matrix, which enables generators in each step. You can imagine a generator is much like the guitarist's finger, but the generator is actually able to play different notes, in fact each generator can play a different note in each step. Now imagine a simple sequence - an empty matrix with just one row and 3 steps, say **Up**, enabled on each step. This would mean that on those steps the **Up** generator is activated and will emit exactly one note, just as a guitarist's finger hitting the string once. And the **Up** generator is playing one note from the chord you are holding, in upwards order. Hence if you hold say C major (C, E, G), the generator is playing C, E, G, C, E, G, ... And of course you may not be holding a chord, but say just a single note, say C. Well, then the sequencer is just repeating C. of course, the sequence can switch between multiple octaves, say, C0, C1, C2, C0..., depending on the **Octaves** control.

Each of the generators works separately and all of them let you create complicated evolving



sequences, which may eventually can generate sequences much longer than the 32 steps. In the example above the simple C major chord contains 3 notes, spread between say 4 octaves gets you 12 notes, and you may enable the **Up** generator in just a single step. That way the generator is triggered just once every sequence and each time it chooses the next note. If you enable all the steps then it will take  $32 \text{ (steps)} * 3 \text{ (notes)} * 4 \text{ (octaves)} = 384$  steps before the sequence starts repeating. If you instead use both **Up/Down** and **Up 2/Down 2** generators, again enabled in just a single step, C major chord, 4 octaves, (I'll spare you the boring mathematics) it will take 4,224 steps before the sequence starts repeating. In the default 1/16x synchronization and 120bpm it makes almost 9 minutes! So the arpeggiator lets you build a fully non-repetitive 9 minute song just by pressing 3 tones!

And it can get much more complicated...



## Random button

Random button generates random settings using the existing presets.



## Presets button

Presets button displays a window where you can load and manage available presets. Hold **Ctrl** when clicking to load a random preset instead.



## Left arrow button

Left arrow button loads the previous preset.



## Right arrow button

Right arrow button loads the next preset.



## Randomize button

Randomize button loads a random preset.



## button

This button copies the settings onto the system clipboard.



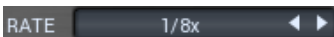
## button

This button loads the settings from the system clipboard.



## Step count

Step count controls the number of steps in the arpeggiator sequencer. Please note that **up/down** and similar generators keep their own order of notes, so for example if you enable **up** for all steps, then it doesn't really matter how many steps you use.



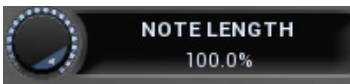
## Rate

Rate defines the sequencer speed. It is always in sync with the host and this parameter essentially controls the speed. The default value of 1/16 means each step takes 1/16th. So in monophonic mode there would be 1 note emitted every 1/16th, 4 times per quarter note.



## Octaves

Octaves controls how many octaves the arpeggiator can produce. 1 octave means only the notes you are holding can be generated. A higher number makes it possible to emit the same notes in higher octaves.



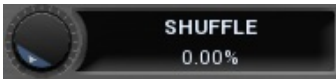
## Note length

Note length controls the length of each note. 100% basically produces legato as each note is directly followed by the next one. Lower values makes the emitted notes shorter, leading to staccato.



## 8x button

8x button increases the note length to be 8x longer. This way the notes may start to overlap. It is useful for arpeggiated pads for example. Please note that because the notes start overlapping, the CPU consumption may rise significantly for some sounds.



## Shuffle

Shuffle modifies the timing so that the arpeggiator stops producing straight notes, but rather shuffle (for a value of 100%) or even dotted (for a value of 200%).



## Sequencer length

Sequencer length controls the maximum sequence length for **up, down...** and other generators. It lets you provide some sense of rhythm.

*For example, if you hold 4 notes and the number of octaves is 3, then 12 notes are generated by say **up** generator before the sequence starts from the beginning. You may want to select a sequencer length of 8, so that it starts repeating every 8 notes, which may be more natural in a typical 4/4 rhythm.*



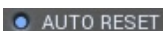
## Monophonic switch

Monophonic switch enables or disables the monophonic mode. This doesn't do anything on its own and affects only the editor behaviour. When enabled, any change in the sequence will ensure that the modified steps contain only one note (generator). If disabled, each step may contain multiple notes (generators) and the output would be polyphonic.



## Ignore velocity switch

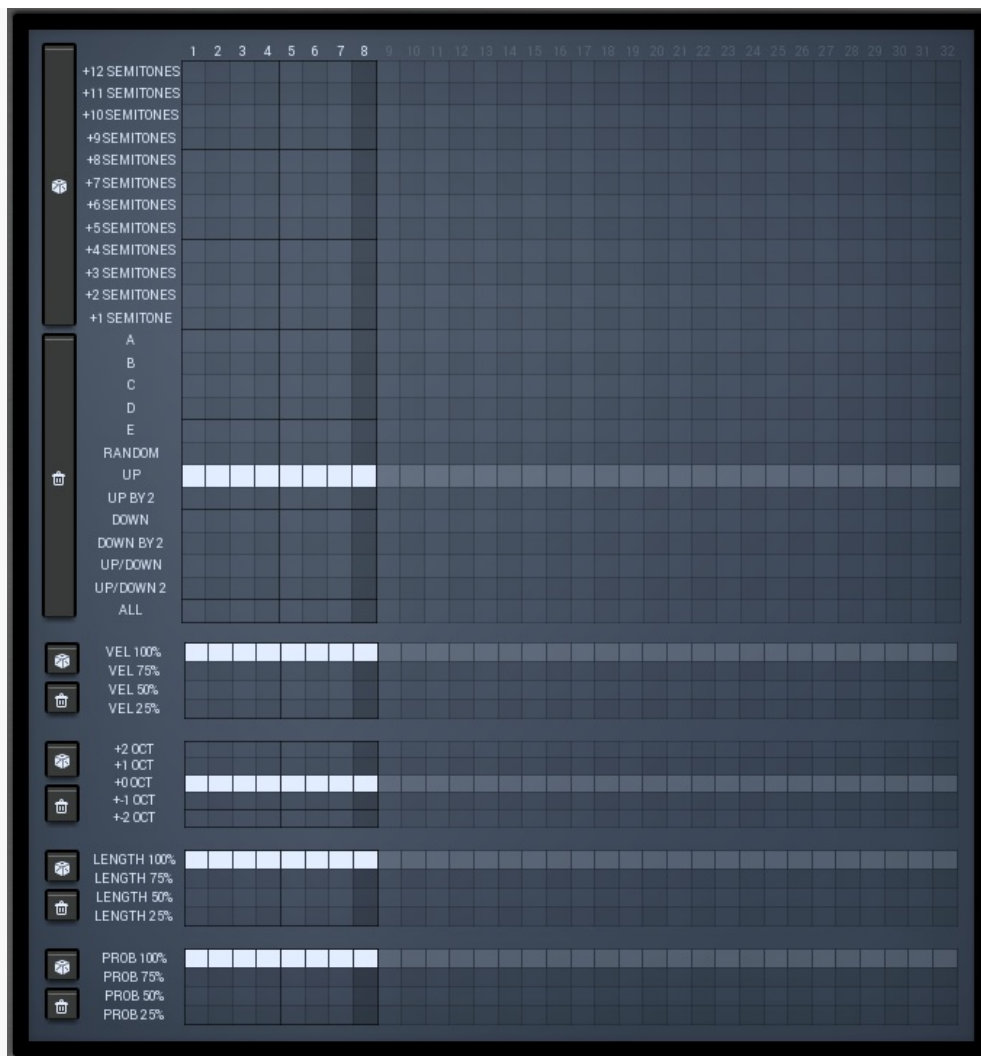
Ignore velocity switch lets the arpeggiator ignore the velocity of the notes that you play. If enabled, the emitted note velocity is defined solely by the velocity matrix. If disabled, the velocity matrix still stands, but the velocity of the input notes still matters.



## Auto-reset switch

Auto-reset switch makes the arpeggiator reset the synchronization every time a note is pressed.

# Sequencer matrix



Sequencer matrix controls which notes are emitted in each step. Each step is defined in one column and each row enables a particular generator in each step. These generators emit exactly one note in each step; the difference is which note is emitted. In polyphonic mode you can enable more than one generator in each step.

**A, B, C, D, E** generators emit the notes you are playing. A is the lowest note, B the one above it etc. Therefore enabling any of these generators will always produce the same note. You can therefore sequence 5 notes freely. If you are holding less than 5 notes, say 3, then D is the same as A and E is the same as B. These generators ignore settings of the **Octaves** parameter, so the played notes only are emitted.

**+n semitones** generators emit the note transposed upwards from the note A. For example if you are holding C major chord (C, E, G), everything is related to C being the lowest note. The +1 semitone generator then produces C#, +2 semitones is D etc. Therefore enabling any of these generators will always produce the same note. If you have selected A, B, C and +9 semitones in 4 steps, you will hear C, E, G and A (which is 9 semitones up from C). These generators ignore settings of the **Octaves** parameter, so the played notes only are emitted. They are displayed above the **A, B, C, D, E** generators to give a good visual representation of the transpositions.

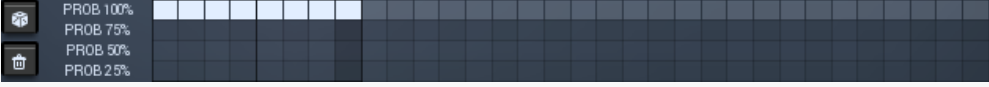
**Up** generator produces sequence starting from the lowest note up, one by one, including higher octaves. For example, if you are holding C major chord (C0, E0, G0) and the **Octaves** is set to 2, then the generator produces the sequence: C0, E0, G0, C1, E1, G1, C0 ... **Up by 2** is similar, but steps by 2, hence the resulting sequence is: C0, G0, E1, E0, C1, G1, C0 ... **Down** generator is the exact opposite of Up and the resulting sequence is: G1, E1, C1, G0, E0, C0, G1 ... **Down by 2** generator is similar, but again steps by 2, hence the resulting sequence is: G1, C1, E0, E1, G0, C0, G1 ...

**Up/down** generator is a basically a merge of Up and Down generators, going from the

lowest note up and then back down, so in the example above the result is: C0, E0, G0, C1, E1, G1, E1, C1, G0, E0, C0... **Up/down 2** generator is similar, but repeats the limit notes, so the sequence is: C0, E0, G0, C1, E1, G1, G1, E1, C1, G0, E0, C0, C0... Combining both Up/down and Up/down 2 generators can produce various evolving sequences.

**Random** generator chooses a random note every time.

**All** generator plays all of the notes every time. It is useful for trance-gate-like effects.

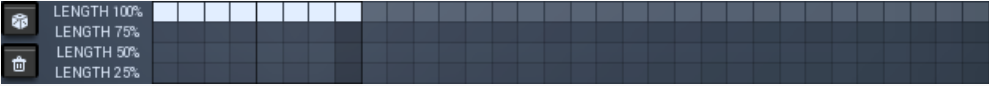


### Probability matrix

Probability matrix lets you adjust the probability that a note in the step will be emitted. The setting affects all notes in the step, so if you are using polyphonic mode, it is not possible to let different notes in a single step have a different probability.

**Randomize button**  
Randomize button generates a random probability matrix.

**Clear button**  
Clear button sets the probability matrix to its defaults.

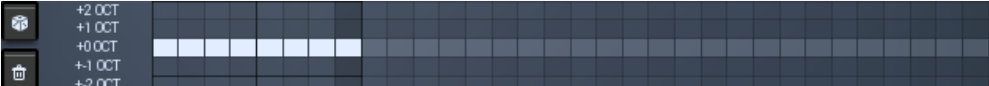


### Length matrix

Length matrix lets you adjust the length of each note. This setting works in conjunction with the global **Length** parameter. The setting affects all notes in the step, so if you are using polyphonic mode, it is not possible for different notes in a single step to have different lengths.

**Randomize button**  
Randomize button generates a random length matrix.

**Clear button**  
Clear button sets the length matrix to its defaults.

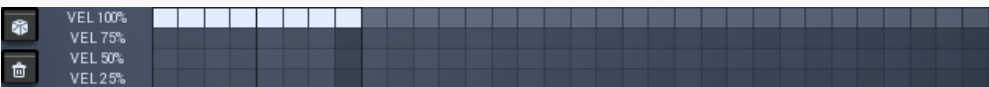


### Octave matrix

Octave matrix lets you transpose all of the notes in each step by a certain number of octaves. The setting affects all notes in the step, so if you are using polyphonic mode, it is not possible to let different notes in a single step be transposed by different number of octaves.

**Randomize button**  
Randomize button generates a random octave matrix.

**Clear button**  
Clear button sets the octave matrix to its defaults.



### Velocity matrix

Velocity matrix controls the velocity of the notes in each step. The setting affects all notes in the step, so if you are using polyphonic mode, it is not possible to let different notes in

a single step have a different velocity.

If **Ignore velocity** is enabled, then the actual velocities of the notes you are holding do not matter and the values in the velocity matrix control the actual velocities of the output notes. These are defined as percentages, but if you are used to the ancient MIDI standard, which uses velocities 0-127, then 100% equals 127, 50% is 63 (or 64) etc.

If **Ignore velocity** is disabled, then the actual velocities are affected by both the velocity of the notes you are holding and the **Velocity matrix**. For example, if you are holding a note with velocity of 60 and the value in the matrix is 50% then the output note velocity will be 30.



### **Randomize button**

Randomize button generates a random velocity matrix.



### **Clear button**

Clear button sets the velocity matrix to its defaults.



### **Randomize button**

Randomize button generates a random sequencer matrix.



### **Clear button**

Clear button sets the sequencer matrix to its defaults.



**button**

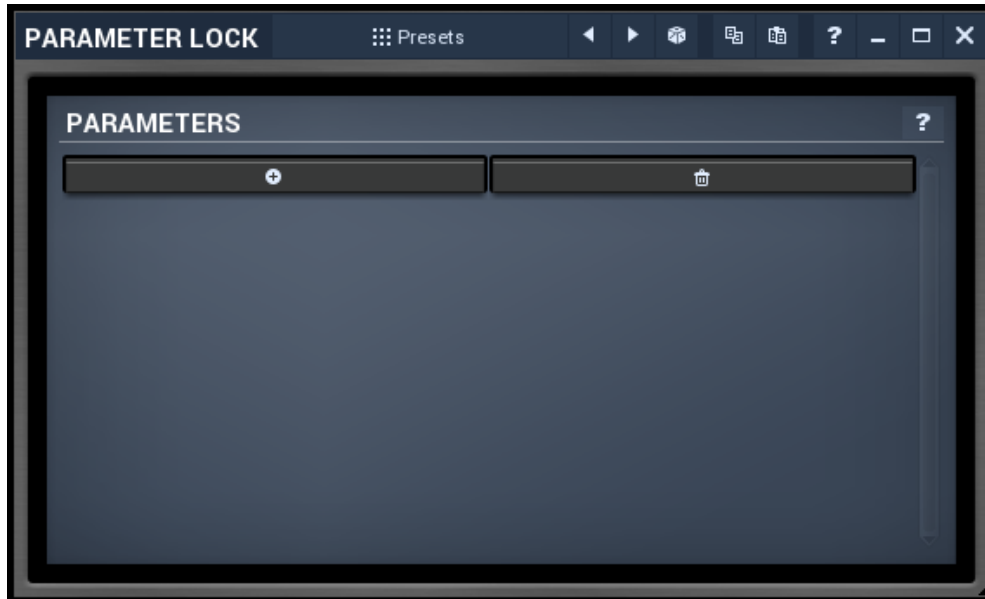
This button shifts the whole sequence one step to the left.



**button**

This button shifts the whole sequence one step to the right.

# Lock feature



Lock feature provides a simple way to keep parameters unchanged when using randomization or browsing presets. You simply use the learn feature (right click) in the same way you would with modulators or multiparameters, and touch every parameter you want to keep locked. You can also select them directly in the parameters lock window where you can save them as presets, copy & paste etc. Note that this is not saved with global plugin presets for obvious reasons.

Presets

## Presets button

Presets button displays a window where you can load and manage available presets. Hold **Ctrl** when clicking to load a random preset instead.

## Left arrow button

Left arrow button loads the previous preset.

## Right arrow button

Right arrow button loads the next preset.

## Randomize button

Randomize button loads a random preset.

## button

This button copies the settings onto the system clipboard.

## button

This button loads the settings from the system clipboard.





## Parameters panel

Parameters panel configures the list of the parameters which are locked.



### button

This button adds a parameter to the list of locked parameters. Alternatively you can use the learn feature available by right-clicking the paramlock button for example.



### button

This button deletes the selected parameter from the list of controlled parameters.

# Multiparameter



Multiparameter is a powerful structure, which can speed up your workflow significantly and even perform automatic tasks, often useful when performing in real-time for example. Essentially a multiparameter is a controller which controls other parameters, in fact, an unlimited number of them. Each parameter has limits and potentially a transformation curve for more advanced processing. By manually moving the multiparameter (or automating/modulating it) you can control all of the associated parameters at once.

*This is just the beginning, but it is worth demonstrating how it could be used. We will show it on a vibrato effect. MMultiBandVibrato (and partly MVibrato) is very good at simulating rotary speakers. A rotary speaker traditionally contains a speed switch, or in our case we will think of it as a speed knob - a control that alters the spin speed of the rotary. This would normally be the **Rate** parameter of the vibrato. However, when the rate is increased, the vibrato starts changing the pitch too much, sounding a little too "honky-tonk". We can compensate for this by lowering the **Depth** parameter. As it is not very convenient to control 2 parameters at once, we use a multiparameter to control both parameters with appropriate ranges (ascending for the **Rate** and descending for the **Depth**).*

Besides this basic usage, multiparameters can also work as triggers and switches. Set a multiparameter's mode to **Trigger** or **Switch** and it stops being a slider and becomes a button. When you click the button, the multiparameter starts moving on its own - over the dialled-in switch time it will increase its value (and also the values of any associated parameters) to a maximum and, in the case of trigger mode, then decrease it back to a minimum. In switch mode clicking the button again, the multiparameter decreases back to the minimum value. To make the multiparameter into a simple switch, we can set the switch time to minimum, but in this case we want to extend the functionality in our rotary example.

*As mentioned, rotary speakers often have a speed switch. Once switched on, the speed starts increasing until it reaches the "fast" setting, and when switched off, the speed starts*

decreasing to the original "slow" rate. All we need to do to replicate this functionality is to set the multiparameter's mode to 'switch'.

A real rotary actually has 2 speakers, one for low frequencies and the other for the higher ones. As you might expect, these do not have the same spin rate nor do they speed up or slow down equally either. Here is where we can start showing the true potential of multiparameters.

To simulate this, we have to use two bands of MMultiBandVibrato, the first one will simulate the lower reproducer, and the second will be the higher. We use the first multiparameter to control the first band's rate in the same way as described in the example above. Similarly, we use the second multiparameter to control the second band's rate. Now we have 2 switches and can make each band speed-up or slow-down separately, but we want just one switch for both bands. To do this, we use a third multiparameter to control the first and second multiparameters, in switch mode again but with a 0ms switch time. Pressing the button of the 3rd multiparameter instantly activates the other 2 multiparameters, they both start speeding-up, over a different time period as we requested. Pressing the button again, releases it which also instantly releases the first 2 multiparameters and they start slowing down. Just like the real thing.

Now that we have shown you what is possible with multiparameters, it is worth mentioning that they are used extensively for building active presets on the easy screens of most Melda plugins. Every multiparameter given a name in the **Information** panel will be shown on the Easy screen (if the plugin has one). Check our online video tutorials to get more information about **multiparameters and about building active presets**.

 Presets

## Presets button

Presets button displays a window where you can load and manage available presets. Hold **Ctrl** when clicking to load a random preset instead.

## Left arrow button

Left arrow button loads the previous preset.

## Right arrow button

Right arrow button loads the next preset.

## Randomize button

Randomize button loads a random preset.

## button

This button copies the settings onto the system clipboard.

## button

This button loads the settings from the system clipboard.

# Behaviour



## **NORMAL SWITCH TRIGGER BANKS METER** Mode

Mode controls the behaviour of the multiparameter.

**Normal** mode makes the multiparameter work like any other control.

**Switch** mode hides the slider and shows a button instead. The button has 2 states. By pushing the button, the multiparameter value starts rising from 0% to 100% over a specified time interval. By pushing it again the value starts falling back to 0%. You could do the same thing having the multiparameter in normal mode and moving the slider from left to right and then back, but mode this performs that automatically and maintains a constant time period.

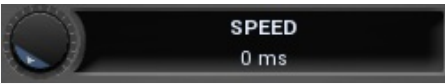
**Trigger** mode is similar to switch mode, but the button has only a single state and when you push it, the value automatically goes from 0% to 100% and then back without any need to push the button again.

**Banks** mode is very different. A multiparameter in banks mode keeps several states (called banks) for all of the parameters, much like A-H presets, but only with a limited set of parameters. The multiparameter then morphs between the banks or can be set to switch directly between them (no interpolated values). This is a marvellous way to control many parameters with complex settings by using a single multiparameter.

Let's explain the banks mode in more detail. Say you switch a multiparameter to banks mode, learn a few parameters and set the number of banks to 4. Then bank 1 contains a value for all of the parameters. Similarly bank 2 contains a different value for each of them. And so on. If you set the multiparameter slider to 0%, the associated parameters will be set to values in bank 1. If you set the slider to 100%, bank 4 will be used. If you set the slider to 33.3%, bank 2 will be used. And what if you select 50%? Then it will be halfway between bank 2 and bank 3.

You can have many banks, you can edit each of them, generate random settings etc. So let's say you want to create some complex movement. You use a multiparameter in banks mode, select a reasonable number of banks. You can edit each of them, but it is easier to use the randomization button to generate random settings for each of them. Then every time you move the multiparameter, all of the associated parameters will move, somewhere between the banks. You can then use a modulator or automation to slowly adjust the multiparameter.

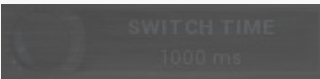
**Meter** mode makes the multiparameter work as a meter. Instead of controlling other parameters it starts following the value of them. You can then use that to implement a simple meter on the easy screen (if the plugin has one).



## Speed

Speed controls the interpolation time. When it is zero and you change the multiparameter value, the associated parameters are adjusted immediately. If this is non-zero however, the actual parameters won't change immediately but will interpolate over time. The speed value is actually the time needed to go from minimum to maximum or vice versa. So if this is 1 second and the current value is say 0% and you click 100%, it will take 1 second for the multiparameter to get there.

This feature is provided mainly because changing some parameter via MIDI or mouse may cause unnecessary zipper noise or inaccuracies due to low MIDI precision. Using the interpolation you can somewhat slow everything down, so that the artifacts become negligible. It can also be used creatively. The default value has been experimentally tested to avoid all artifacts for most parameters.



## Switch time

Switch time defines the time needed to switch from the minimum value to the maximum one, or conversely. It is used only in **switch** and **trigger** modes.



## Value mode

Value mode defines the units displayed on the multiparameter. **Percents** mode lets the multiparameter display percentages between 0% to 100%. **By first parameter** mode uses the current value of the first parameter that is controlled by the multiparameter. For example, if you want to control a plugin gain, but also in addition to the changed gain control other parameters, you may still want to call the multiparameter "gain" and the units should be decibels as usual, not percentages which do not make much sense for such a multiparameter.

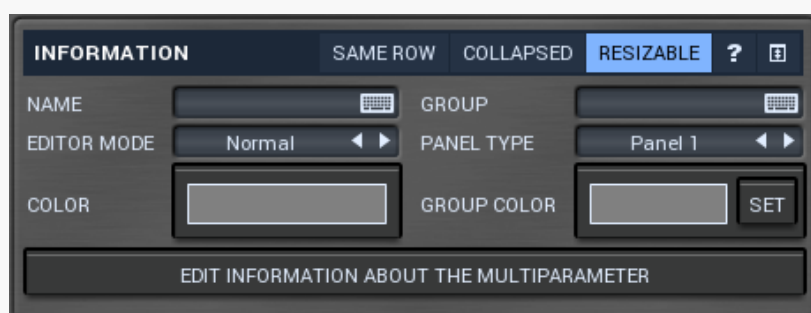


## Set the current value as default

### default

Set the current value as default stores the current value as the default one for the multiparameter. The multiparameter's control responds to right-click by setting the default value in the same way that other parameters do. This way you can select the default value. It is also essential when building your own active-presets.

# Information



SAME ROW

## Same row

Same row checkbox defines if the parameter should be displayed next to the previous one on the Easy screen. Otherwise it will be placed on the next row. This setting serves as a hint and the plugin may ignore it, if it is impossible to do.

COLLAPSED

## Collapsed

Collapsed checkbox lets you specify if the panel should be collapsed by default. This is ignored for all multiparameters except for the first one in a group.

RESIZABLE

## Resizable

Resizable switch lets you specify if the panel could be resized. It is on by default to make sure everything gets resized, however when using multiple panels next to each other, it may be advantageous to disable resizing of some of them to save space. Otherwise each panel's size is proportional to number of controls it contains, which could make some of the panels larger than actually necessary.

NAME

## Name

Name specifies the name of the multiparameter, which is shown on the multiparameter button. The name is also used for active presets - if the name is specified, then the multiparameter serves as a parameter for the active preset (on the Easy screen). If no name is specified or the first character is \*, then the parameter is hidden. This is useful if you need some internal multiparameters which you don't want to show on the Easy screen for some reason.

GROUP

## Group

Group can be used to put some multiparameters into the same group, which results in them being placed in the same panel on the Easy screen (the active preset editor).

EDITOR MODE

Normal

## Editor mode

Editor mode controls the way the multiparameter are to be displayed on the Easy screen.

**Normal** is the default mode and is represented by a small knob or button.

**Big** mode is similar, but uses a big knob or big button.

**Checkboxes** makes the multiparameter displayed as a set of checkboxes (also called radio buttons). It is relevant only in **Banks** mode.

**Title button** places the control into the title bar of the panel to which it belongs.

**Title enable button** places the control into the title bar of the panel to which it belongs and makes it a standard enable button (which also makes all controls within the panel unavailable if it is itself disabled).

**XY pad** creates a 2 dimensional XY pad control, that edits this multiparameter in the X axis and the next multiparameter in the Y axis.

**Meter** creates a simple meter instead. You will probably want to set the multiparameter to Meter mode as well or attach it to a modulator. Meters don't really control anything and their purpose is purely to get a visual feedback. The meters can be horizontal or vertical and they can be up or down. Up is the usual choice useful for peak meters for example. Down is useful for gain reduction meters.

PANEL TYPE

Panel 1

## Panel type

Panel type defines the type of panel in which multiple controls of the same group are placed. These differ only in their graphics display.

A dark grey rectangular UI element with the word "COLOR" in white text on the left and a white square color selection box on the right.

## Color

Color defines colorization for the element on the Easy screen (if the plugin has one). The feature is disabled if the Alpha value of the color is 0. Using this feature often increases memory consumption of the plugin, so make sure you use it only if necessary and try to use as low a number of different colors as possible. It is recommended to use only the snapshot colors to make sure the same colors are used in most cases, reducing the memory consumption. It is also highly recommended to use colors with a value (lightness) of 128 (the middle value), which makes sure that the lightness of the elements won't be changed. This works best for most styles. Please note that the style may be configured to simply ignore this color, so there may be no change at all. If you use this feature, make sure that you test it with all styles.

For the sake of workflow the colors have predefined meanings. It's highly recommended to follow this standard:

Blue - modulation

Green - equalization, filtering

Brown/yellow - reverb, delay

Orange - dynamics

Red - limiting, saturation, distortion

Cyan/yellow - stereo

Purple/pink - time, pitch, unison...

Grey - utilities, tools

A dark grey rectangular UI element with the text "GROUP COLOR" on the left, a white square color selection box in the middle, and a "SET" button on the right.

## Group color

Group color defines colorization for the group panel on the Easy screen (if the plugin has one) and is ignored for all multiparameters except for the first one in a group. The feature is disabled if the Alpha value of the color is 0. Using this feature often increases memory consumption of the plugin, so make sure you use it only if necessary and try to use as low number of different colors as possible. It is recommended to use only the snapshot colors to make sure the same colors are used in most cases, reducing the memory consumption. It is also highly recommended to use colors with a value (lightness) of 128 (the middle value), which makes sure that the lightness of the elements won't be changed. This works best for most styles. Please note that the style may be configured to simply ignore this color, so there may be no change at all. If you use this feature, make sure you test it with all styles.

For the sake of workflow the colors have predefined meaning. It's highly recommended to follow this standard:

Blue - modulation

Green - equalization, filtering

Brown/yellow - reverb, delay

Orange - dynamics

Red - limiting, saturation, distortion

Cyan/yellow - stereo

Purple/pink - time, pitch, unison...

Grey - utilities, tools

A small dark grey square button with the word "SET" in white text.

## Set button

Set button sets the color and group color for all multiparameters in the same group. It is



pretty sensible to do that as all controls should look similar within each group. This can also be done by editing each parameter, but this way is easier.

# Parameters panel



Parameters panel configures how the multiparameter assigns values to the target parameters.



## button

This button adds a parameter to the list of controlled parameters. Alternatively you can use the learn feature available by right-clicking the multiparameter button.



## button

This button deletes the selected parameter from the list of controlled parameters.

PARAMETER

GLOBALS - GAIN

## Parameter

Parameter defines the target parameter which is being modulated. The set contains all automatable parameters.

RANGE MODE

Interval

## Range mode

Range mode defines how the parameter range is selected. While sometimes it is better to specify minimum and maximum, other times it is better to use a nominal center and depth (% of full scale). This control allows you to define which one it will be.

**Up and down** mode makes the values go above and below the selected **Value**, which is considered the center. The interval is made smaller if necessary.

**Full range mode** is similar, except the range is symmetrically constrained, so the selected **Value** may not be the center anymore.

**Up/down only modes** goes from the selected value up/down only.

Let's compare these 4 modes. Taking a value of -12dB value, with a depth of 75% and a

scale of +/- 24dB. The nominal range is therefore = +/-24 dB \* 75% = 36dB. With values of 0%, 50% and 100% the outputs are:

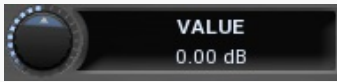
Up and down: -24, -12, 0 (range constrained to 12 dB either side)

Full range: -24, -6, 12 (range limited to minimum, but not constrained)

Up only: -12, 6, 24 (range not constrained = +/-24 dB \* 75% = 36dB)

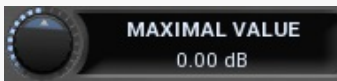
Down only: -12, -18, -24 (range limited to minimum)

**Interval mode** is the most simple one and goes from **Value** to **Maximal value**.



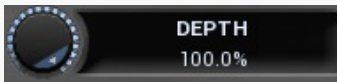
### Value

Value defines the center of the target parameter's range or the minimum if the **Range mode** is set to **Interval**.



### Maximal value

Maximal value defines the upper limit of the target parameter's range. It is available only if the **Range mode** is set to **Interval**. This value can be lower than **Value**. 0% is always mapped to reference>Value and 100% to reference>Maximal value.



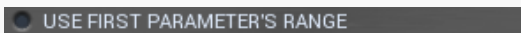
### Depth

Depth defines size of the target parameter's range. It is used only if the **Range mode** is not set to **Interval**.



### Invert

Invert checkbox inverts the target parameter's range, so that minimum becomes maximum and vice versa.



### Use first parameter's range

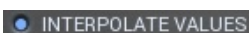
Use first parameter's range makes the parameter display use the same range as the first parameter in the list. This is often useful if want to control the range in some way and apply the range to multiple parameters.



### Show

## transformation shape button

Show transformation shape button displays the graph editor, which lets you tweak the shape of the curve used to control the selected parameter. The X axis shows the original values, the Y axis defines the results. Please note that this takes some CPU, therefore you have to enable it using the enable button in the title bar.



### Interpolate values

Interpolate values controls if the parameter value is to be interpolated between the bank values or if it will take the value from the nearest bank. For example, when bank A contains the value 0% for the parameter and bank B contains 100% and you set the multiparameter to 30%, then when interpolation is enabled, 30% is selected for that parameter, when the interpolation is disabled, the nearest value, 0%, is selected. If you want the parameter to step from one bank value to another then disable interpolate values.



### Set to all parameters buttons

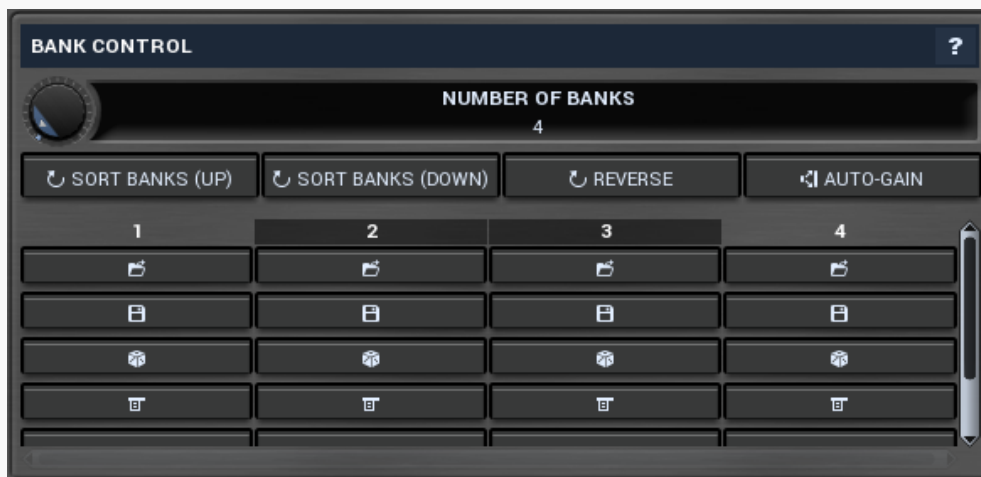
Set to all parameters buttons sets the interpolate values setting for all parameters controlled by that multiparameter.

● CYCLIC MODE

## Cyclic mode

Cyclic mode switches the multiparameter into so-called cyclic mode. If you have say 4 banks, called A, B, C and D, and gradually increase the multiparameter value, it starts with A, then interpolates to B, then to C and finally to D. But after that you cannot interpolate back to A, because D is the last one, the maximum value. In cyclic mode the multiparameter behaves as if there were a clone of A at the end, hence after D is reached, the multiparameter interpolates back to A and creates a full circle A->B->C->D->A. This is handy for example if you use a saw wave modulator to drive the multiparameter and want to repeat the sequence of the banks.

# Bank control panel



Bank control panel is available only in **Banks mode** and contains tools to define the banks between which the multiparameter is interpolating. The multiparameter stores parameter values for each bank. Here you can load and save these values.

Each bank has 5 buttons and a value for each controlled parameter. Click the **load button** to load the bank values into the plug-in. If you want to change say bank C, you first click its **load button**, change whatever you need and resave the settings.

By clicking the **save button** you overwrite the bank's settings from those currently set in the plug-in. A typical approach to define the multiparameter's behaviour is to set the number of banks, then go to the plugin editor, set all associated parameters to the values you would like to have in bank A and click the save button for bank A, then modify the parameters to whatever you want in bank B and click the save button for bank B, etc. You can also use the **Random button** to generate random values using the smart-randomization engine for each of the banks.

And the **menu button** enables you to re-order the banks

For each bank, the values for each parameter are shown and can be changed as desired.



**Number of**

## banks

Number of banks controls the number of settings that the multiparameter stores for all parameters. By changing the multiparameter value all associated parameters are then modified according to these settings. Please note that when you change the number of banks, the multiparameter will behave differently, because the multiparameter's range from 0% to 100% will now be distributed between a different number of presets. If you had automated the multiparameter value in your host for example you will almost certainly need to edit / rewrite the automation envelope.

 SORT BANKS (UP)

### Sort banks (up) button

Sort banks (up) button reorders the banks so that the values of the selected parameter are in increasing order.

 SORT BANKS (DOWN)

### Sort banks (down) button

Sort banks (down) button reorders the banks so that the values of the selected parameter are in decreasing order.

 REVERSE

### Reverse button

Reverse button reverses the order of banks, so that the first bank contains values of the previously last one and so on.

 AUTO-GAIN

### Auto-gain button

Auto-gain button temporarily enables AGC and automatically sets up the main plugin gain to each bank so that all banks provide similar output loudness. To use it, ensure that the main gain parameter is attached to the multiparameter, start playback of your sound material then click the button. and press this button. It will take several seconds to complete depending on the number of the banks.



### button

This button loads the bank settings by setting all associated parameters to the values in the particular bank.



### button

This button saves the current values of all associated parameters into the particular bank. So you can edit all those parameters in the plugin then click the save button to store them in the bank.



### Randomize button

Randomize button loads random settings to the bank using the smart randomization engine. Only parameters associated with the multiparameter are randomized.

Generally, randomization in plug-ins works by selecting random values for all parameters, but rarely achieves satisfactory results, as the more parameters that change the more likely one will cause an unwanted effect. Our plugins employ a smart randomization engine that learns which settings are suitable for randomization (using the existing presets) and so is much more likely to create successful changes.

In addition, there are some mouse modifiers that assist this process. The smart randomization engine is used by default if no modifier keys are held.

Holding **Ctrl** while clicking the button constrains the randomization engine so that parameters are only modified slightly rather than completely randomized. This is suitable to create small variations of existing interesting settings.

Holding **Alt** while clicking the button will force the engine to use full randomization, which sets random values for all reasonable automatable parameters. This can often result in "extreme" settings. Please note that some parameters cannot be randomized this way.

Hold **Shift** while clicking the button to undo the previous randomization.



### **button**

This button provides some additional options related to the bank.



### **Name button**

Name button lets you rename the bank.

# MIDI settings window



MIDI settings window lets you configure, how the plugin reacts to various MIDI messages. You can use MIDI controllers or MIDI notes and you can also configure a controller to switch between presets, which is especially useful for realtime performances.

Presets

## Presets button

Presets button displays a window where you can load and manage available presets. Hold **Ctrl** when clicking to load a random preset instead.

## Left arrow button

Left arrow button loads the previous preset.

## Right arrow button

Right arrow button loads the next preset.

## Randomize button

Randomize button loads a random preset.

## button

This button copies the settings onto the system clipboard.

## button

This button loads the settings from the system clipboard.

CONTROLLERS

NOTES

PRESET SWITCH

## Tab selector

Tab selector switches between subsections.

# Controllers panel



Controllers panel contains settings of MIDI controllers.

DO NOT LOAD FROM PRESETS

## Do not load from presets button

Do not load from presets button disables loading the controllers from presets. This may be handy if you have configured specific MIDI controllers with target parameters and you want to browse the presets without the need to configure them every time. Please note that some presets may rely on specific controllers though. For example, if a preset requires a velocity controller to provide velocity-dependent response, this option will avoid loading it, so the preset won't be complete, until you reconfigure it.

LAST NOTE-ON CHANNEL ONLY

## Last note-on channel only button

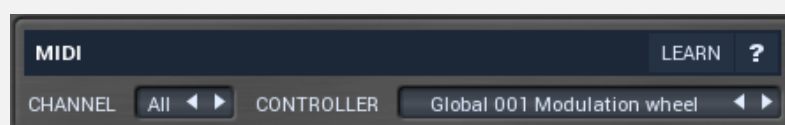
Last note-on channel only button makes the engine more suitable for voice-per-channel devices. These devices are able to send different controllers for each note you press, which however means that these could collide. This option makes the engine pass only the controllers that are related to the last note you pressed. For classic keyboards it is not relevant as you will usually use a single MIDI channel to transmit both the controllers and notes. Some more modern keyboard controllers will allow you to select one MIDI channel for the notes and a different one (or the same one) for the controllers.

GLOBALS - GAIN

## Parameter button

Parameter button lets you choose the parameter being controlled. The set contains all automatable parameters.

# MIDI



LEARN

## Learn

Learn enables or disables MIDI learn.

CHANNEL

All

## Channel



Channel defines the controller MIDI channel.

CONTROLLER Global 001 Modulation wheel ◀ ▶ **Controller**

Controller defines the source controller.

## Values



RANGE MODE Interval ▶ **Range mode**

Range mode defines how the parameter range is selected. While sometimes it is better to specify minimum and maximum, other times it is better to use a nominal center and depth (% of full scale). This control allows you to define which one it will be.

**Up and down** mode makes the values go above and below the selected **Value**, which is considered the center. The interval is made smaller if necessary.

**Full range mode** is similar, except the range is symmetrically constrained, so the selected **Value** may not be the center anymore.

**Up/down only modes** goes from the selected value up/down only.

Let's compare these 4 modes. Taking a value of -12dB value, with a depth of 75% and a scale of +/- 24dB. The nominal range is therefore = +/-24 dB \* 75% = 36dB.

With values of 0%, 50% and 100% the outputs are:

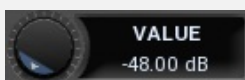
Up and down: -24, -12, 0 (range constrained to 12 dB either side)

Full range: -24, -6, 12 (range limited to minimum, but not constrained)

Up only: -12, 6, 24 (range not constrained = +/-24 dB \* 75% = 36dB)

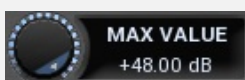
Down only: -12, -18, -24 (range limited to minimum)

**Interval mode** is the most simple one and goes from **Value** to **Maximal value**.



### Value

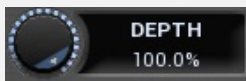
Value defines the center of the target parameter's range or the minimum if the **Range mode** is set to **Interval**.



### Maximal value

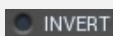
Maximal value defines the upper limit of the target parameter's range. It is available only if the **Range mode** is set to **Interval**. This value can be lower than **Value**. 0%

is always mapped to reference>Value and 100% to reference>Maximal value.



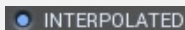
## Depth

Depth defines size of the target parameter's range. It is used only if the **Range mode** is not set to **Interval**.



## Invert

Invert checkbox inverts the controller shape, so the minimum becomes the maximum etc.



## Interpolated

Interpolated makes the controller value interpolated over the time using the smart interpolation. This approach ensures there won't be abrupt changes, which could lead to clicks and pops. However sometimes you may want to apply these changes immediately - for example when changing ADSR based on the note velocity, in which case this parameter should be disabled.

# Notes panel



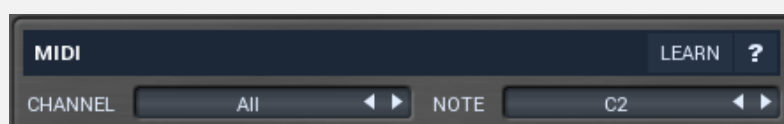
Notes panel contains settings of MIDI note controllers, if you want to control parameters using MIDI keys.

GLOBALS - GAIN

## Learn

Learn enables or disables MIDI learn.

# MIDI



CHANNEL All **Channel**

Channel defines controller MIDI channel.

NOTE C2 **Note**

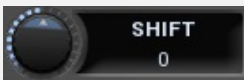
Note defines the controller's target MIDI note. It is used only in On/off mode.

## Values



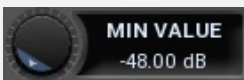
MODE Logarithmic **Mode**

Mode controls how the controller works. **Logarithmic** scale is useful for oscillator frequencies, however it may not be useful for general parameters. **On/off** modes react only to single notes and can be used for triggers. **Switch** modes are similar, but only recognize when a note is pressed. In all octaves mode it doesn't matter which octave is used. So this is useful, when you for example want use any note C to switch something on and off.



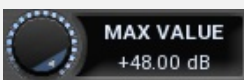
### Shift

Shift lets you shift the original note up or down by the specified number of semitones.



### Min value

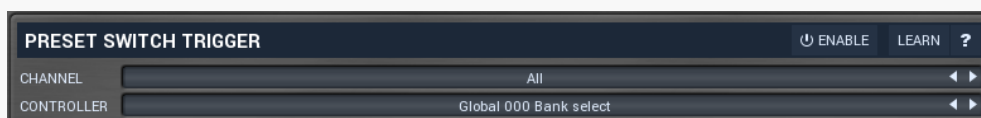
Min value defines the minimum value for the target parameter.



### Max value

Max value defines the maximum value for the target parameter.

## Preset switch trigger panel



Preset switch trigger panel lets you select a MIDI controller, which will switch presets. It

provides the same action as clicking the arrows next to the main preset button. When the controller value gets below 33%, the previous preset is loaded. When the controller value gets above 66%, the next preset is loaded.

## LEARN **Learn**

Learn enables or disables MIDI learn.

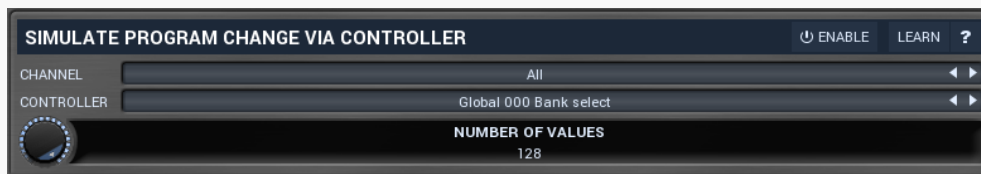
CHANNEL  **Channel**

Channel defines controller MIDI channel.

CONTROLLER  **Controller**

Controller defines source controller.

# Simulate program change via controller panel



Simulate program change via controller panel lets you select a MIDI controller, that will work as program change, for convenience. You can use it then to switch between A-H presets or presets via panel below.

## LEARN **Learn**


Learn enables or disables MIDI learn.

CHANNEL  **Channel**

Channel defines controller MIDI channel.

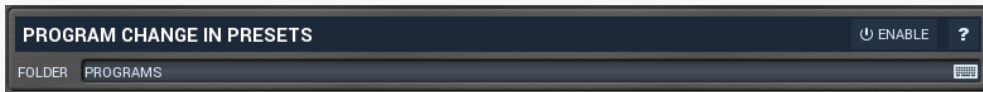
CONTROLLER  **Controller**

Controller defines source controller.

 **Number of values**

Number of values defines the number of programs to switch between. By default Program change MIDI standard offers 128 programs. However it may be too many and could be hard to actually control with the specific controller. Hence you can lower the number of actual programs.

# Program change in presets panel



Program change in presets panel enables the MIDI program change processing. If disabled, the plugin follows Program Change messages by changing the A-H presets. The obvious disadvantage is that this way there are just 8 presets. By enabling this feature the plugin stops selecting A-H presets and rather loads different presets from the specified preset folder, including all sub-folders. The default folder is called "Programs". To use it, you simply need to create a preset folder called Programs and put the presets into it. Note that the order matters of course. And you can change the folder name at any time, so you can have several sets of selectable presets.

FOLDER PROGRAMS

 **Folder**

Folder defines the preset folder from which the presets for program-change MIDI messages are taken.

# Used controls

Here we discuss the general properties of all application controls. As a most important rule you should note, that you can always use any question mark button or F1 (or Ctrl+F1 or Ctrl+H) key with the mouse cursor over a specified control to get detailed information about what it does and how to use it.

## File selector



File selector contains a directory tree that you use to move across the file system. In many cases the range is limited to a sub-tree to maintain better organisation. Next to the tree you can see the list of files present in the selected directory. There is a splitter between both of them that you can use to change the width of both of the lists.

## Zoomer



Zoomer provides a simple way to zoom and move in an enlargeable view.

- **Plus button** zooms-in.
- **Minus button** zooms-out.
- **Zoom default button** zooms to the default ratio, which typically means full zoom-out.
- **Lock button** zooms to the default ratio, which typically means full zoom-out.

## Tab-set



Tab-set is typically used wherever there is too much to edit, but not enough space to display it all. It can be also used to switch between possible alternatives.

- **Left mouse button** selects a tab.
- **Ctrl + Left mouse button or Right mouse button** displays the whole tab in a pop-up window (this is not used for all sets of tabs). This comes handy when you want to have multiple tabs visible at the same time.
- **Left and Right arrows** select the neighbouring tab.

- Click on one of the buttons on the border to scroll the control and show tabs that are currently invisible.

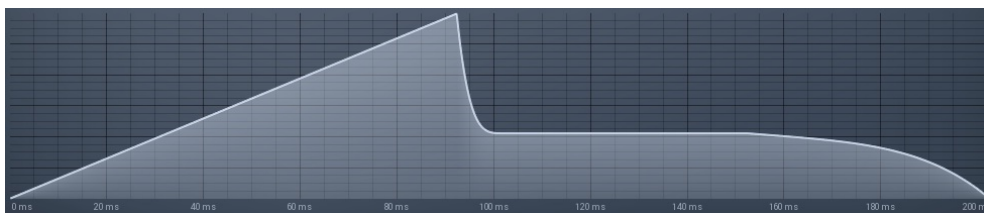
## Value button



Value button is an alternative to the tracker and its main advantage is that it is very small. In some cases the button simply serves as a clickable item and a menu is shown when clicked. However the mouse wheel and other controls still do work.

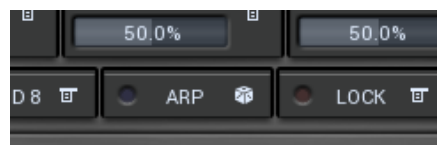
- **Click and drag using the left mouse button** to change the value.
- **Right mouse button** selects the default value.
- **Mouse wheel, arrow keys** and vertical drag using **middle mouse button** or using **left mouse button while holding Ctrl** modifies the value more precisely.
- **Home key** configures the minimal possible value, conversely **end key** setups the maximal one.
- **Esc or Backspace keys** restore the original value when either one is pressed during dragging.
- **Shift + left mouse button** or **double-click using left mouse button** lets you edit the value as text. You can use the virtual keyboard or type on your computer keyboard. In some cases this shows a menu with all possible values instead.
- **Alt + press then release** measures the time between the press and the release and applies it as time/frequency tap. Usable only for certain values of course.

## Graph editor



Graph editor will show and edit one or more graphs.

## Listbox

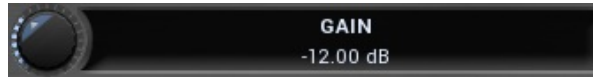


- **Left mouse button** selects an item. If you click on a check-box, the item will be checked/unchecked.



- **Mouse-wheel** scrolls the view.
- **Arrow keys, Page Up** and **Page Down** move the selection.
- **Space** checks or unchecks the selected item(s).
- **Home** key selects the first item, **End** selects the last one.

## Knob



Knob is an alternative to a tracker, which simulates physical knobs.

- **Click and drag using the left mouse button** to change the value.
- **Right mouse button** selects the default value.
- **Mouse wheel, arrow keys** and vertical drag using **middle mouse button** or using **left mouse button while holding Ctrl** modifies the value more precisely.
- **Home key** configures the minimal possible value, conversely **end key** setups the maximal one.
- **Esc or Backspace keys** restore the original value when either one is pressed during dragging.
- **Shift + left mouse button** or **double-click using left mouse button** lets you edit the value as text. You can use the virtual keyboard or type on your computer keyboard. In some cases this shows a menu with all possible values instead.
- **Alt + press then release** measures the time between the press and the release and applies it as time/frequency tap. Usable only for certain values of course.

## Tree-control



- **Left mouse button** selects an item. If you click on a check-box, the item will be checked/unchecked.
- **Mouse-wheel, Up** and **Down** arrow keys, **Page Up** and **Page Down** move the selection.
- **Enter** opens/closes the selected item.
- **Space** checks or unchecks the selected item.
- **Home** key selects the topmost item, **End** selects the last one.

# Switcher



Switcher is an alternative to a tracker or knob control, but it has a limited set of values.

- **Left mouse button** shows a menu with list of all possible values. This function might be unavailable in certain cases when the number of possible values is too high.
- **Up** and **Down** arrow keys, **buttons** in the control and **mouse-wheel** increase or decrease the value.

## About MeldaProduction

The best sound on the market, incredible workflow and versatility beyond your imagination. We create the deepest and the most powerful audio plugins with unbelievable sound and tons of unique features you cannot find anywhere else.

## Innovative Thinking

At MeldaProduction, we make the most advanced tools for music production and audio processing. We get inspired by the whole range of tools from the ancient analog gear to the newest digital creations, but we always push forward.

We've always felt the audio industry is extremely conservative, still relying on the prehistoric equipment making the job unnecessarily slow and complicated. That's why we invent new technologies, which make audio processing easier, faster, better sounding and more creative.

## Sound Matters

In the world full of audiophiles you just need superb audio quality. And that's why we spend so much time perfecting audio algorithms until they sound unbeatable. Everything from dynamic filters to spectral dynamic processing. Our technologies just sound perfect.

## Inspiring User Interface

Modern user interfaces must not only be easy and quick to use, but also versatile and the whole visual appearance should inspire you. MeldaProduction plugins provide the most advanced GUI engine on the market. It is still the first and only GUI engine, which is freely

resizable and stylable. Our plugins can look as an ancient vintage gear, if you are working on old-school rock music. Or as super-modern futuristic devices if you are working on modern electronic music.

## Easy to Use, Yet Versatile

The only limit is your imagination. Our plugins are with absolutely no doubt the most powerful and versatile tools on the market. Yet we managed to make the plugins easy to use via the active presets and smart randomization system. But when you are ready, you are one click away from the endless potential the plugins provide.

## Never-Ending Improvements

Most companies create a plugin, sell it and abandon it. We improve our plugins, add features, optimize... until there is nothing left to improve and there are no more ideas. Unfortunately that hasn't happened yet :). And the best thing is that the updates are free-for-life!

*MeldaProduction was founded in 2009 by Vojtech Meluzin and is based in Prague, Czech Republic.*