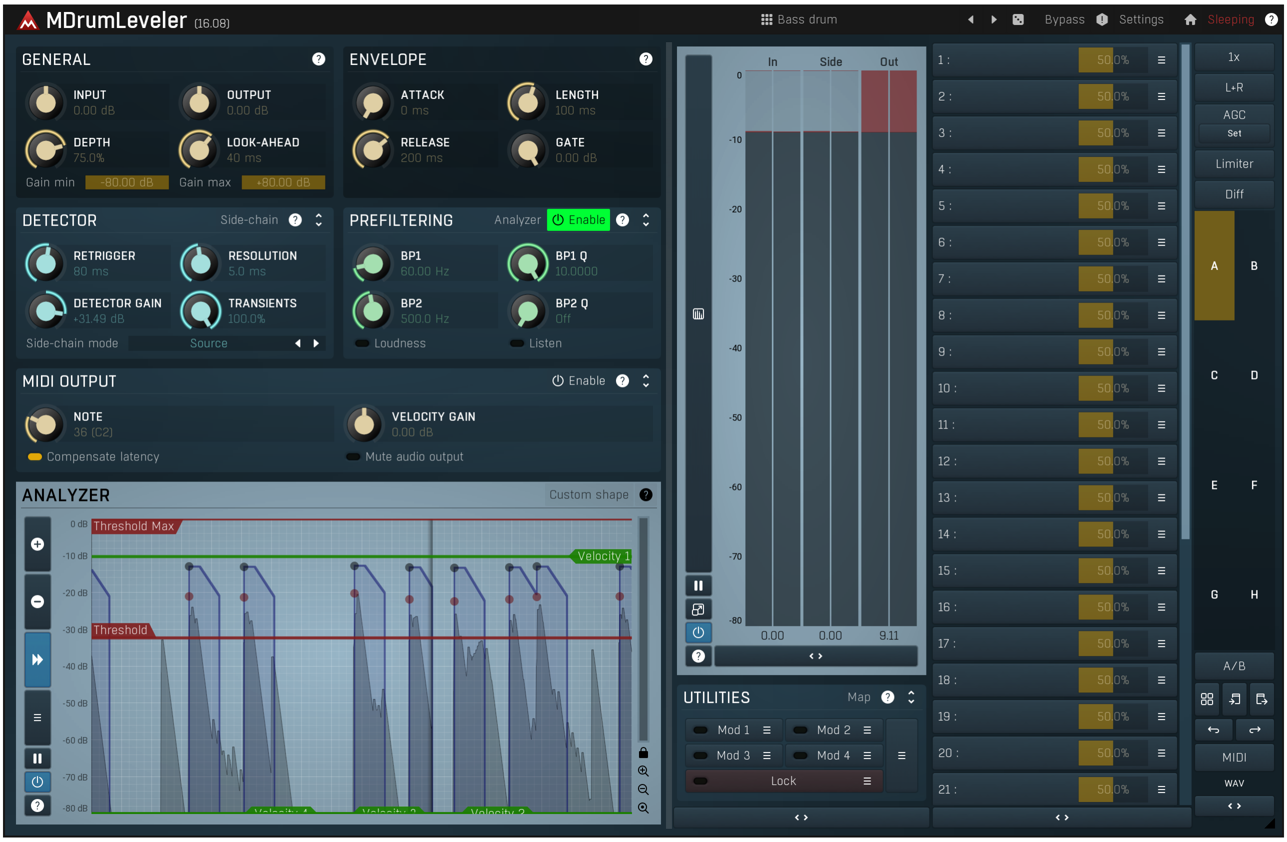
MDrumLeveler



MDrumLeveler (16.08)

GENERAL
INPUT 0.00 dB
OUTPUT 0.00 dB
DEPTH 75.0%
LOOK-AHEAD 40 ms
Gain min -80.00 dB
Gain max +80.00 dB

ENVELOPE
ATTACK 0 ms
LENGTH 100 ms
RELEASE 200 ms
GATE 0.00 dB

DETECTOR
RETRIGGER 80 ms
RESOLUTION 5.0 ms
DETECTOR GAIN +31.49 dB
TRANSIENTS 100.0%
Side-chain mode Source

PREFILTERING
BP1 60.00 Hz
BP1 Q 10.0000
BP2 500.0 Hz
BP2 Q Off

MIDI OUTPUT
NOTE 36 (C2)
VELOCITY GAIN 0.00 dB

MDrumLeveler is a unique dynamics processor specifically designed for percussive materials, primarily drums, but it can be used on any rhythmical audio. Similarly to compressor and expanders it manipulates level, but it works in quite a different manner. Traditional dynamic processors detect levels of input samples and transform them depending on the transformation curve. This means that the gain is changing on every sample. This can be very well used to shape the sound itself, but it is hard to make it transparent if, for example, you want to level drum hits.

MDrumLeveler on the other hand analyzes the incoming audio and detects events (drum hits for example) in it. It then detects the level of each event and transforms it in some way. The typical use is to even out drum hits. No drummer's performance is perfect, each hit has a different velocity, and differences in levels are even exaggerated by the recording equipment. But there is a very high standard for perfection in performances for most modern styles. Vocalists regularly use automatic tuner, drummers have MDrumLeveler now.

Basics of MDrumLeveler

MDrumLeveler is very easy to use, however for some specific materials, especially those with lots of leakage ("bleed") between drums the work can get problematic. But it's different from what you know about compressors for example, so we need to start from the basics.

Event detection

Most of the settings can be made within the MDrumLeveler's analyzer. The first stage of the processing is event detection. An event can be anything from a drum hit to a guitar strum. The detector is pre-processing the signal to to make it easier to distinguish these events. The "detector signal" is displayed as a **black graph** in the analyzer and usually has the typical shape of a series of hills, each of which can be an event.

Your first task is to set the red **Threshold marker**. Any hill reaching above the threshold will be considered an event. There is also a **Threshold Max marker**, which controls the maximum level (height of the hill), but you'll rarely need this one.

Please note that the following assumes that the factory default colours are used for the graphs; they can all be changed, saved as presets etc.

For every event the plugin draw 2 dots, a red one and a black one. The **Red dot** indicates the original (input) event level and the **Black dot** is the calculated output event level. Note that these levels are not related to the black graph at all. The detector signal (black graph) is used only for event detection and calculation of the event levels is a very different story. You don't need to think about it much, just don't be concerned that the levels of dots and the black graph don't match.

The plugin tries to put output events at the levels indicated by the green **Velocity markers** and this is your second important thing to setup. In most cases you'll be fine with just 1 velocity marker, the rest of them left in the bottom making them disabled. If the **Depth** is 100%, then each black dot (output event level) will be placed exactly at a velocity marker, so you could say that "every hit will have the exact same velocity". That may result in quite a robotic performance, but since the level is not at all the only property of a drum hit, it can actually sound quite natural, just "played by a perfect drummer". The lower the Depth, the less change towards the velocity marker is made, the black dots will be somewhere between the red dots (original level) and the velocity marker (perfect level).

If there is just one velocity marker, the "perfect" level is obviously defined by this marker. But what if there is more than one? Then the selected marker is the closest one above the event's red dot, or just the highest one if there are no more markers above it. But why would you even use multiple markers? For example the lower marker can be for ghost notes and the higher marker for accented notes.

What happens to the events

So now you have the events detected using the threshold markers, the event levels are computed and the output levels are correctly selected using the velocity markers. What happens to the events then? Gain, that's all, like with all dynamic processors. But the important thing is that the gain is constant for the length of the event, so that there is no coloration of the events like it would be with compressors for example. The drum hits will sound the same as before, just with different level. If you want it to be transparent of course.

For each detected event an attack-length-release envelope is started. **Attack** is usually 0ms, because the plugin smartly detects the beginning of each event, so that there is no need for a fade-in. **Length** controls the time the gain stays the same and should cover the main body of the drum hit. Finally the **Release** stage brings the gain back to normal. Use these parameters to shape the sound. If you want it to be transparent, use longer Length and Release.

There's one more interesting parameter - **Gate**. This controls the "normal" gain applied between the events. This provides an exceptionally accurate and smart gate you can use to remove the ambience and leakage from other microphones for example. And for this you may actually not use any velocity markers, so that the plugin will only be a gate and won't change the levels of the events at all.

The typical scenario

You simply want to improve the performance by making the hits similar in level and remove the leakage from other instruments, so you have the "pure" track. That's necessary for modern tight mixes and very hard to accomplish with standard tools such as compressors and gates. So here's the plan:

1. Insert the plugin on the track you are processing.
2. Set the threshold marker so that only the events you want to process are detected. If you are not sure, set Gate to -40dB, so that the plugin kills everything except for the detected events.
3. Fix the envelope parameters if needed. First set Gate to -40dB, so the settings are easier to hear. In most cases the defaults should work just fine, but for example hi-hat may need shorter **Length or Release** or both, otherwise bleed from other drums will start kicking in after the hit. Conversely toms may need longer **Length or Release** or both, otherwise the plugin may kill the hits before they actually end and that may sound unnatural. Set the Gate back to 0dB when finished.
4. Set the velocity marker(s) and Depth. This depends on what you want, but let's start with the simple scenario when the drummer's performance is not to be very dynamic, which is a typical case for contemporary music. So set Depth to 100% for starters and just use one velocity layer and place it about at the highest red dot level. Then if you move it up, the plugin starts working kind of like an expander highlighting the events against the background noise and leakage (that's usually desired). Conversely when you put it lower, it will start compressing the loud events. Feel free to play with the velocity layers, there is no good or bad setting. When finished, set the Depth so that it sounds good and still natural (if you aim for a natural sound of course).
5. Set the Gate. The events should now be processed well, so it's just a question of what to do with the ambience and leakage between the them. For acoustic music you may need to keep it at 0dB, as the ambience is part of these styles. For contemporary music you may need as clean and tight sound as possible, so just move the Gate down until it sounds the way you want. Typically -6dB to -10dB seems fine.

Targeting the processor to specific drum hits

When your drum kit is well recorded without much leakage, the job is easy. But recording drums is generally the most problematic exactly because it is never perfect. And to make the plugin work well, you need to make it detect the right hits. So let's go through some typical scenarios.

Bass drum

Bass drum is usually easy enough. A microphone is placed from the back (not the beater), usually inside. That avoids most leakage. If there is some, use a **band-pass** filter in the (**Prefiltering** panel) with some pretty low frequency, say 50Hz and rather a low Q. Most other drums are not emitting anything below 100Hz, so this should do the trick just fine. If you need it, you can use a higher Q, but it may resonate too slowly, so the detector signal may be too lazy. Therefore always try to use lower Q if possible.

If you used some nonstandard approach or used several microphones, e.g. one from the front, which can pick-up a lot of snare sound, make at least one of the microphones work well and use this side-chain trick for those which cannot be fixed : All of the microphones are recording the same performance, so first use MAutoAlign to get them in phase (you should do that anyway) and then send the "good" microphone to the side-chains of the MDrumLeveler plugins on all the "bad" ones and enable the side-chain processing by pressing the **Side-chain** button, so that the bad mics follow the good one. The drummer's performance is the same and since it is working for the "good" microphone, it should now work for the "bad" ones as well.

Hi-hat

Hi-hat is at the exact opposite on the spectrum scale - it takes the highest frequencies. The problem is, other drums take that too. Fortunately the levels of the hi-hat hits are quite high, so again you can just use a BP filter at say 8kHz and a reasonable Q.

Hi-hat is problematic, because it can be closed (short) or open (long). If the drummer usually plays a closed hi-hat and you aim for short crisp sound, use shorter Attack and Release, otherwise you will have to find some compromise with longer values.

Toms

Tom microphones often record lots of leakage from literally everything. The good thing is that they are not played that much, so in the worst case you can automate some parameters, e.g. threshold, but also that the toms are often quite "tuned". Actually a good practice is to tune the toms to the scale of the song, but not many musicians do that probably. Anyway, the fact they are tuned makes it possible to use a band-pass filter and find the exact frequency of the drum.

Try a sonogram (MAnalyzer, MAutoDynamicEq and all our equalizers etc.), it makes it easy to spot the frequency easily. Classic analyzers don't work that well, because all of the drum hits everywhere filling the spectrum. The sweet spot is usually between 100Hz and 200Hz. Then use a higher Q (the lower the better but...) so you can set the threshold appropriately. The toms usually don't have a very quick transient anyway, so smoothing it even more with a resonant band-pass filter doesn't harm that much. If it does, try lowering the **Transients** parameter to make the plugin focus less on the transients that are not there anymore. It can especially help with leakage from the snare drum.

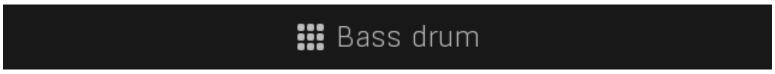
In extreme cases a bass drum (or snare drum) can interfere a lot with the toms and from brief listening to the unprocessed track you may be able to tell that the leakage is just too high. There's still a hope using the following trick. If you expect a tight track without any leakage (so lots of gating) and you assume the drummer is not playing the problematic drum (let's say the bass drum) at the same moment as the tom, then you may use a side-chain compressor before the plugin and send the bass drum to its sidechain with very low threshold and high ratio. This will simply kill the tom track on every bass drum hit, so that MDrumLeveler doesn't even notice it.

Snare drum

Snare drum is probably the most problematic yet the most important (along with the bass drum). There is usually lots of leakage from the hi-hat and just about any other part of the drum kit. It is often recorded using multiple microphones from top, bottom and in rare cases even side, and it is often played in many ways - normal hits and rim shots for example. And of course, they need to sound good.

Again we need to remove leakage. Snare drums may be tuned, but unless the snares are loosened, there is no specific pitch other than what the metal rim produces and that's rarely useful. So try finding some frequency that is present a lot in the drum hit, but not in other parts of the kit. A sonogram can be a huge help here again. The spot will usually be somewhere between 200Hz and 600Hz. If the drummer uses a rim shot, that one may have a completely different pitch defined by the drum, the stick and even the position where the back end of the stick was touching the drum head. But there usually is some form of pitch if the drummer is good enough, so you can use the second band-pass filter to target this frequency as well. The filters are in parallel, so this way you can make one plugin instance tuned to both the normal hits and the rim shots.

In most cases the energy from the snare drum is much higher than from other drums, but if the leakage is bad and the drummer is playing dynamically, you might be in trouble. In the worst case scenario it may be impossible - for example if there are 2 snare drums with similar sound and the drummer is playing dynamically, it will be impossible to distinguish loud hit on one drum from soft hit on the other one. But you may try the side-chain compressor trick from the tom section, or you may just need to lower the amount of processing - less gating and lower depth. If the snare drum is recorded using multiple microphones, you may use a "good" one to drive the "bad" ones using the side-chain method explained in the bass drum section.



Presets

Presets button shows a window with all available presets. A preset can be loaded from the preset window by double-clicking on it, selecting via the buttons or by using 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. The arrow buttons next to the preset button can be used to switch between presets easily.

Holding **Ctrl** while pressing the button loads a random preset. There must be some presets for this feature to work of course.

Presets can be backed up by 3 different methods:
A) Using "Backup" and "Restore" buttons in each preset window, which produces a single archive of all presets on the computer.
B) Using "Export/Import" buttons, which export a single folder of presets for one plugin.
C) By saving the actual preset files, which are found in the following directories (not recommended):
Windows: C:\Users\{username}\AppData\Roaming\MeldaProduction
Mac OS X: /Library/Application support/MeldaProduction

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

Please note that prior to version 16 a different format was used and the naming was "{pluginname}presets.xml". *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 feature is manually maintained in order to remove generally unusable presets, so it may take some time before any 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

Left arrow button loads the previous preset.

Right arrow

Right arrow button loads the next preset.

Randomize

Randomize button loads a random preset.

Panic

Panic 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

Settings

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

Licence manager lets you activate/deactivate the plugins and manage subscriptions. While you can simply drag & drop a licence file onto the plugin, in some cases there may be a faster way. For instance, you can enter your user account name and password and the plugin will do all the activating for you.

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.

Global system settings contains some settings for all MeldaProduction plugins. Once you change any of them, restart your DAW if needed, and it will affect all MeldaProduction plugins.

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 zippering noise, but more CPU will be used.

WWW

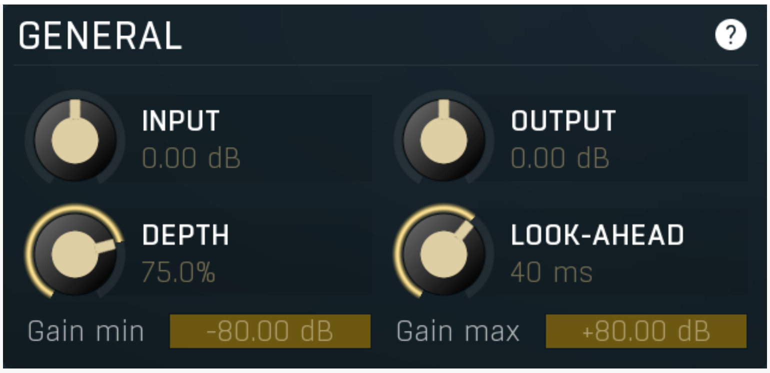
WWW button shows a menu with additional information about the plugin. You can check for updates, get easy access to support, MeldaProduction web page, video tutorials, Facebook/Twitter/YouTube channels and more.

Sleeping

Sleep indicator

Sleep indicator informs whether the plugin is currently active or in sleep mode. The plugin can automatically switch itself off to save CPU, when there is no input signal and the plugin knows it cannot produce any signal on its own and it generally makes sense. You can disable this in Settings / **Intelligent sleep on silence** both for individual instances and globally for all plugins on the system.

General panel



INPUT 0.00 dB

OUTPUT 0.00 dB

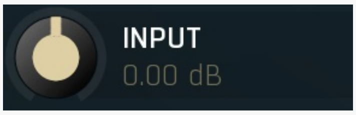
DEPTH 75.0%

LOOK-AHEAD 40 ms

Gain min -80.00 dB

Gain max +80.00 dB

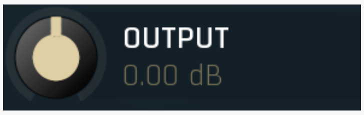
General panel contains the main parameters controlling the plugin behaviour. Please pay especially high attention to the **Depth** parameter.



INPUT 0.00 dB

Input gain

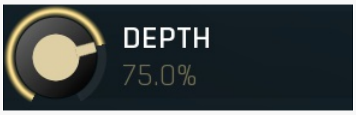
Input gain defines the power modification applied to the input signal. You may need it if the input level is too high or too low for the detector to work properly or which would require you to set the threshold too high or low as that would simply be inconvenient. Note however there is a dedicated **Detector gain** parameter as well, which doesn't affect the actual audio signal at all.
Range: -24.00 dB to +24.00 dB, default 0.00 dB



OUTPUT 0.00 dB

Output gain

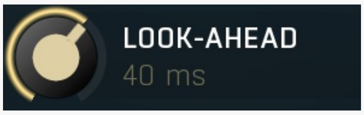
Output gain defines the power modification applied to the output signal. Since the plugin essentially changes the level of the signal, or rather individual hits, you may need this to fix the output level.
Range: -24.00 dB to +24.00 dB, default 0.00 dB



DEPTH 75.0%

Depth

Depth controls how much of the effect is applied and can be thought of as dry/wet control.
For example, if the detector finds an event with level say -30dB and your nearest velocity marker is at say -20dB, the plugin aims to perform a gain of +10dB to this event. That way the output level of the event would be -20dB. This could make the performance 'too good', almost robotic, so you may want to lower this value. In this example, 100% sensitivity caused +10dB gain to that event. 50% would make it +5dB (to -25dB), and 0% would make it +0dB, so the plugin would do nothing, leaving it at -30dB.
Range: 0.00% to 100.0%, default 50.0%



LOOK-AHEAD 40 ms

Look-ahead

Look-ahead controls how far ahead in time the plugin looks. Since nothing can really look into the future, this in effect causes a delay, which is reported to your host as latency, so a properly functional host would compensate for it and it would cause no problems during mixing or mastering. However you may need to address it when using the plugin real-time. The bigger the look-ahead is, the higher the latency of course.
The further it can look into the future, the bigger the chance that it finds the highest peak of each event. With 0ms look-ahead the plugin can still detect events, but it cannot know their level, because when it detects them, the input level is still very low. But if it can look into the future, then it can wait for the highest peak.
So how much should you set? If you look at a drum hit in a wave editor, the typical time between the start of the hit and the moment with maximum level (the length of the initial 'hill') is usually less than 4ms, but with bass drums it can often reach 10ms. If you are using it real-time, you want the delay to be as low as possible. It is recommended to keep it at least 2ms. A good approach is to set **Depth** to 100%, use one single velocity at maximum level and let the drummer play really badly :). This way the plugin will have lots of work to do. You then just increase the look-ahead until the output starts sounding pretty 'robotic', which is a good indicator that the level measurement is working well.
Range: 0 ms to 100 ms, default 20 ms

Gain min -80.00 dB

Gain min

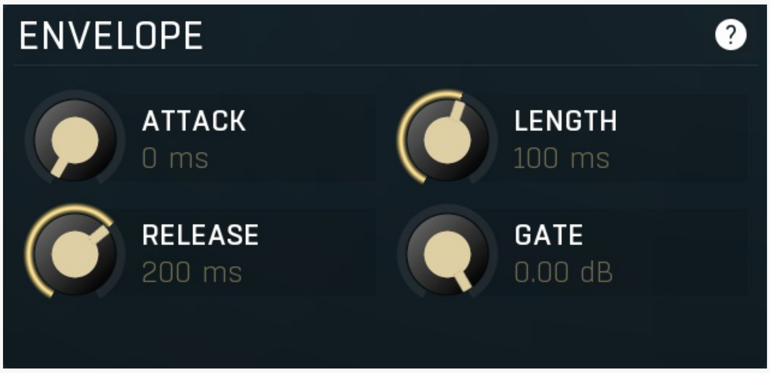
Gain min controls the minimum gain that the plugin can perform on each event. You can use it to limit the event processing. For example, if you don't want the plugin to attenuate the events more than -20dB, then this would be your target value.
Range: -80.00 dB to 0.00 dB, default -80.00 dB

Gain max +80.00 dB

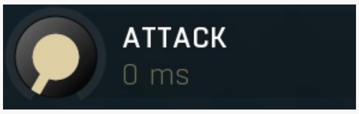
Gain max

Gain max controls the maximum gain the plugin can perform on each event. You can use it to limit the even processing. For example, if you don't want the plugin to amplify the events more than +20dB, then this is your target value.
Range: 0.00 dB to +80.00 dB, default +80.00 dB

Envelope panel



Envelope panel contains the parameters controlling the shape of the events. When the detector find an event and computes its correct level, the processor needs to apply the requested gain to the event. To do that it starts an envelope, which initially increases the gain during the **Attack** stage, then keeps the gain at the target value for time of the event **Length** and then decreases it back to 0dB change in the **Release** stage. This ensures the processing sounds natural, smooth and fixes the performance. You may change these parameters to adjust it to your audio material or even for some creative processing.



Attack

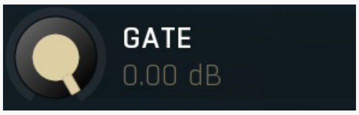
Attack controls the length of the attack stage. You rarely need to touch this parameter, because the processor usually detects the beginning of each event very precisely, so that the transition is still smooth. If you hear clicks at the beginning of the events, the detector is not working properly (which often occurs with too low look-ahead) and you may need to adjust the attack.
Range: 0 ms to 50 ms, default 0 ms

Length

Length controls the length of the event during which the maximum (or minimum) gain is applied. This should cover the body of the drum hit, which can be anything from 2ms with hihats to 400ms with toms. You may want to lower it provide a sort of gate effect and remove the unwanted ambience, leakage from other instruments etc.
Range: 0 ms to 1000 ms, default 50 ms

Release

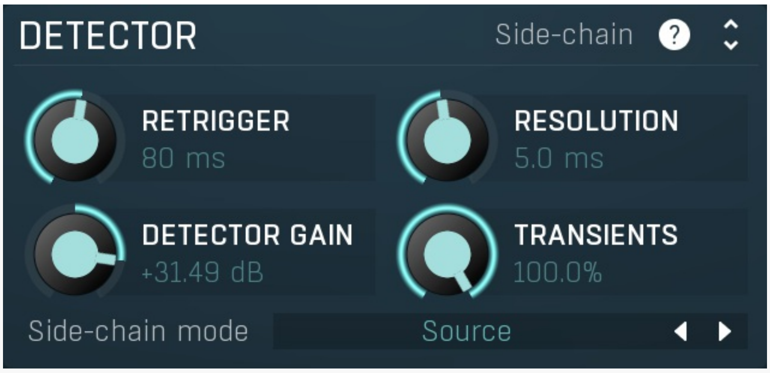
Release controls the length of the release stage during which the gain is coming back to 0dB change. This works hand in hand with the **Length** parameter and has a similar effect. Note however that if you set this too low, an abrupt end may occur after each hit, which would sound as a click. It may be used creatively of course.
Range: 0 ms to 1000 ms, default 150 ms



Gate

Gate controls the level between the events. This gives you an opportunity to create an extremely accurate event-based gate useful mainly to remove leakage from other instruments and to shape the actual drum sound.
Range: -40.00 dB to 0.00 dB, default 0.00 dB

Detector panel



DETECTOR

Side-chain

RETRIGGER 80 ms

RESOLUTION 5.0 ms

DETECTOR GAIN +31.49 dB

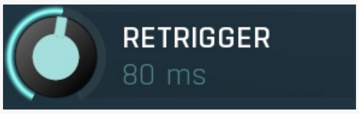
TRANSIENTS 100.0%

Side-chain mode: Source

Detector panel contains some advanced detector parameters controlling how the events are detected.

Side-chain

Side-chain button activates the side-chain input. When enabled, the processor doesn't analyze the regular input but uses the secondary input instead. It can be used for some creative purposes, e.g. rhythmizing a pad using drums, or to fix some specific materials. For example, you may record a snare drum using 2 microphones, first one from above and second one from below. The first one is very close to the hihat, so there is a huge leakage. The bottom one isn't close to any other instrument, so you can process it well enough. But since both microphones record the same performance, you may use the second microphone as a control signal to process the first one as well. And you do that simply by sending the second microphone to the side-chain input.

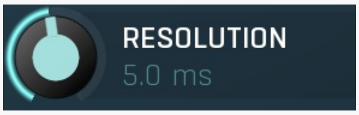


RETRIGGER 80 ms

Retrigger

Retrigger defines the minimum time between 2 successive events. You may want to increase it if the detector finds more events in a row then there actually are. Many drummers for example play in too stiff a way, pressing the sticks against the drum head, which often makes the stick jump away and hit the head again. This creates 2 hits close to each other and to avoid processing both of them you may need to adjust the retrigger time.

Range: 1.0 ms to 1000 ms, default 50 ms



RESOLUTION 5.0 ms

Resolution

Resolution controls the internal detection window time. Change this parameter only if you have a problem adjusting the detector for your audio material.

Range: 0 ms to 100 ms, default 5.0 ms

DETECTOR GAIN +31.49 dB

Detector gain

Detector gain defines the power modification applied to the detector signal, displayed as a black graph. You can use it if the input signal is too high or low, for example when pre-filtering with high resonance caused the level to drop too much. It would be too inconvenient to use the global **Input gain** for it, so this is an easier method which affects only the detector signal, the actual audio is not changed at all.

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

TRANSIENTS 100.0%

Transients

Transients controls how much the detector uses the information about transients in the signal. In most cases you will leave the default value of 100%, which makes the detector fully employ the transient detection. However in certain situations with lots of leakage you may need to lower this value. A typical example is processing a tom microphone with lots of snare drum leakage. While snare drums have very sharp and strong initial transient, toms usually don't and so with the default 100% value the detector tends to produce higher snare levels than tom levels even though it's not a snare microphone. In that case you may want to lower the value and the detector starts to prefer longer transient hits such as toms.

Range: 0.00% to 100.0%, default 100.0%

Side-chain mode: Source

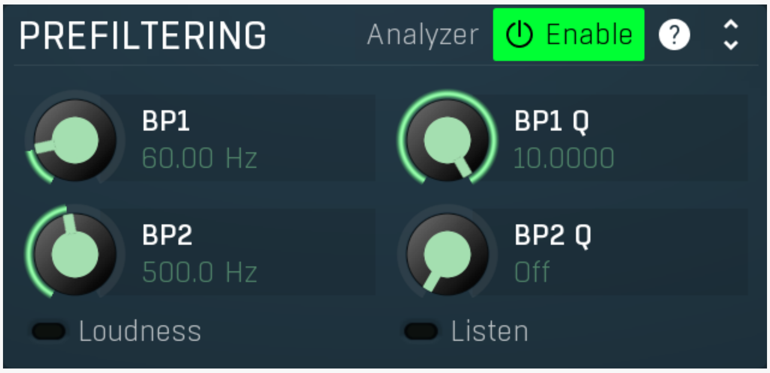
Sidechain mode

Sidechain mode controls the way sidechain is used when activated. **Source** is the default mode and means that the sidechain is used instead of the main input for detection. This is useful when for example you want to use one bass drum microphone signal to drive another one, which contains too much leakage.

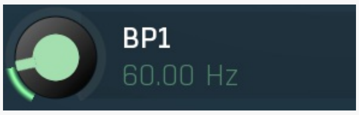
Remove from input uses the sidechain signal to attenuate the detector signal, so that the events are detected only if the sidechain signal is silent enough. For example if you have a significant leakage of bass drum into a snare drum microphone, you can send the bass drum microphone to the sidechain and the plugin will basically remove the bass drum hits.

Present in both combines both input and sidechain signals for the detector, so that the events are detected only if there is enough signal in the sidechain. For example, if you record a snare drum from top and bottom using 2 microphones, the microphones may pick up different leakage, bottom is recording bass drum and the top takes toms for example. By sending one microphone to the sidechain of the other one, the detector detects events only if there is signal in both of them, which minimizes the chance that it would create an event from a leaked tom hit, unless there was a bass drum hit at the same time.

Prefiltering panel

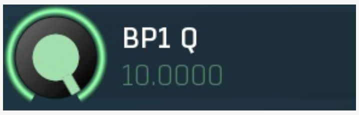


Prefiltering panel contains some filtering parameters that let you preprocess the detection signal. There are mainly 2 parallel band-pass filters, which let you specifically target 2 resonant frequencies in the signal and may come handy if you for example process a signal with lots of leakage, e.g. a hihat microphone which has recorded lots of leakage from the snare drum.



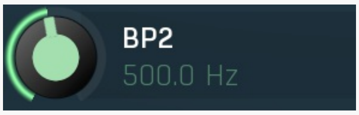
BP 1

BP 1 defines the band-pass filter center frequency used to preprocess the detector signal. This can be used to tune the detector to a specific drum frequency. Please note that the filters are run in parallel, so you can target multiple frequencies.
Range: 20.00 Hz to 20.0 kHz, default 250.0 Hz



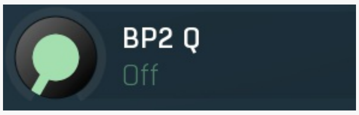
BP1 Q

BP1 Q controls the resonance of the band-bass filter. The higher the value, the more resonant the filter is and the more tuned it is to the center frequency. Set this to minimum to disable the filter. Please note that if the resonance is too high, the detector signal will start losing the initial transient so you may want to check the **Transients** parameter.
Range: Off to 10.0000, default Off



BP 2

BP 2 defines the band-pass filter center frequency used to preprocess the detector signal. This can be used to tune the detector to a specific drum frequency. Please note that the filters are run in parallel, so you can target multiple frequencies.
Range: 20.00 Hz to 20.0 kHz, default 500.0 Hz



BP2 Q

BP2 Q controls the resonance of the band-bass filter. The higher the value is, the more resonant the filter is and the more tuned it is to the center frequency. Set this to minimum to disable the filter. Please note that if the resonance is too high, the detector signal will start losing the initial transient so you may want to check the **Transients** parameter.
Range: Off to 10.0000, default Off

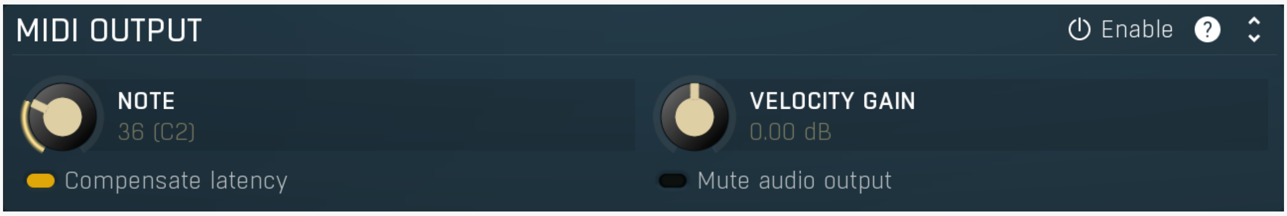
Loudness

Loudness button enables EBU based loudness filter. Since human hearing is quite different from the actual signal measurements, you may use it if you feel the detector levels are not quite similar to what you actually hear.

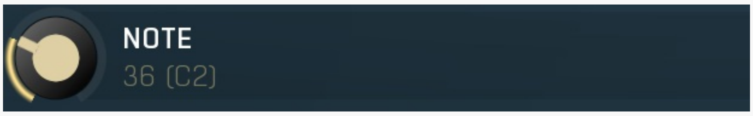
Listen

Listen lets you audition just the filtered signal, so that you can adjust the filters easily. The entire metering system will still work as normal at that moment.

MIDI output panel



MIDI output panel contains parameters of the optional MIDI output.



Note

Note defines which note should be transmitted for every event. By default it is 'off', meaning that there are no notes produced by the plugin.
Range: 0 to 127, default 36

Velocity gain

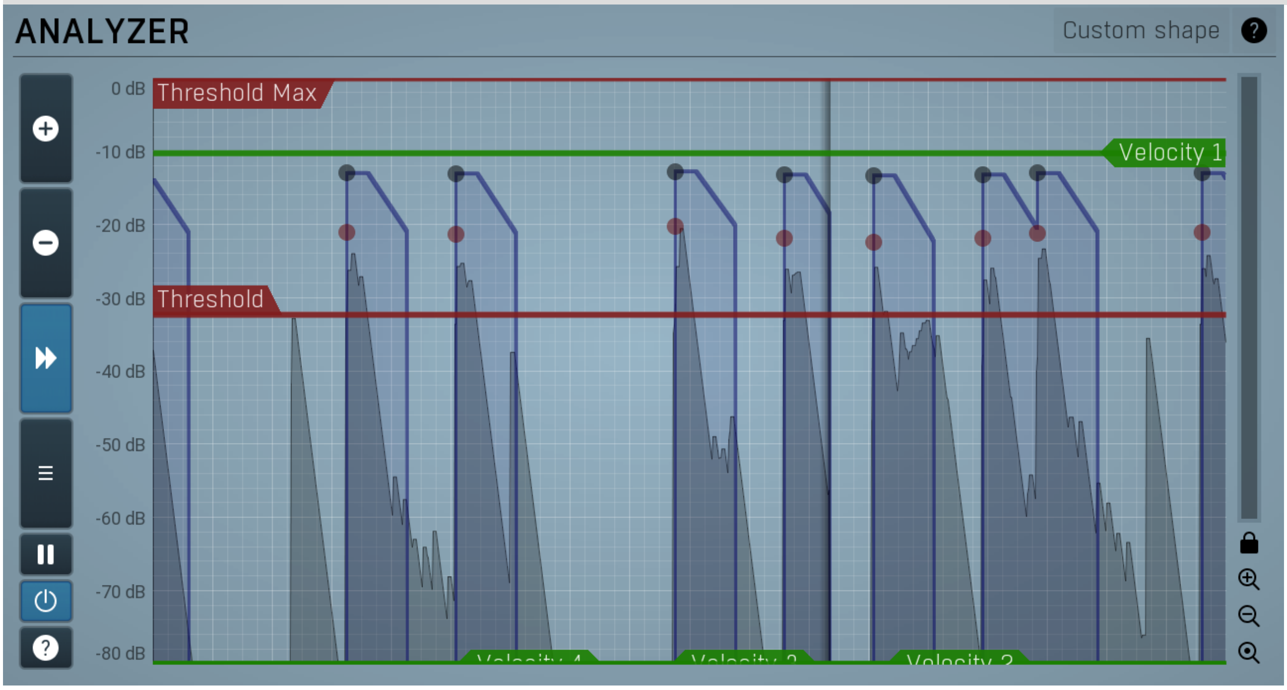
Velocity gain lets you increase or decrease the level of the detected hits, which in effect transforms the output velocity.
Range: -40.00 dB to +40.00 dB, default 0.00 dB

Compensate latency

Compensate latency enables latency compensation for notes generate into the MIDI output. It is provided, because some DAWs compensate latency themselves, but others do not.

Mute audio output

Mute audio output mutes the audio output when the MIDI output is enabled. It is useful in DAWs, where placing a plugin after another automatically sends MIDI from the first to the second, so you can easily put MDrumLeveler in front of a drum sampler for example.



Analyzer

Analyzer displays a continual analysis of the input signal and lets you control the most important parameters easily.

There are **red and black dots** (please note that this description assumes that the default colour palette is used for the time-graph and the background is light, the colors may vary for different styles). The red dots represent detected events above the threshold with their input levels. For every red dot, there is a corresponding black dot placed at the event's output level. In other words, red dots are the input events, black dots are the output events. The vertical position defines the event level, (sort of loudness of the hit).

The black graph displays the **Input detector level**. It will most likely like look like a sequence of hills. Each hill represents a potential event which may be processed and tags a transient in the input signal including its detected level. Please note that these levels are NOT the same as levels of the events (red dots) - while an event level is basically its loudness, the detector signal is pre-processed in a specific way that makes it easy to recognize individual events and set the threshold properly. As such this graph is highly dependent on the detector settings, which you will usually keep at default, but if the graph looks strangely unpredictable, then you might need to adjust the detector settings or input pre-filtering.

The red **Threshold marker** controls the input threshold, the minimum level of a 'hill' to be considered an event. You should set it up so that the hills of events that you want to detect are above it and the hills below it are ignored. In most cases the detector pre-processes the levels (the black graph) extremely well and there will be a significant gap between the real events and the background rumble and leakage from other instruments, so all you need to do is to put the threshold slightly below the level of the smallest wanted event.

The red **Threshold Max marker** controls the maximum input threshold, the maximum level of each event. Events with levels above this threshold are ignored. It can be useful to process events only falling within a certain level range, e.g. only ghost notes, without affecting the loud hits. The **threshold max** marker is disabled when set to maximum (0dB).

There are 4 green **Velocity markers**, which control the desired output levels. If a marker is in the bottom of the analyzer (at -80dB), it is disabled and ignored. If there is no active marker, the processor does nothing unless **Gate** is used. Let's now assume Depth is 100%, maximum processing, which will probably result in quite a robotic performance. For each input event (red dots) the plugin selects a velocity marker, which defines the desired output level for that event. So there will be one black dot for each red dot and each of these black dots will lie on a velocity marker. The lower the Depth is, the further away the black dots can be from the velocity markers.
The velocity markers are essentially exactly what the name implies. They mark velocities. For example, in most pop and heavy music there are no 'ghost' notes on the snare drum and more or less everything sounds accented. So there is just one velocity level and the plugin will try to fix the drummer's performance and assume he wanted to play the same velocity every time. If you assume the drummer plays more dynamically however, you may use multiple velocity markers, say one for silent ghost notes, one for normal hits and one for ultra-high level accents.
This is how it works in detail: When an event is started, the plugin first chooses which marker this belongs to. If there is just one, then the choice is obvious. If there are more than one, then the marker selected is the one closest above it, or just the highest one if there are no more markers above it. Then the plugin selects the output level for the event, which is simply the level of the marker.
If there are no markers enabled, the plugin basically does nothing, unless you use gate. With one single marker you control the levels of all events above the threshold. If the marker is low, then the plugin will basically compress them, if the marker is high, the plugin will expand them. With multiple markers the situation gets a little more interesting and the plugin will basically be placing the events into predefined levels. But the highest marker controls only the output level, because there are no markers above it, so you can basically again compress or expand the loudest hits.

The **blue graph** shows the actual envelopes of the processed events. It serves mainly as a feedback when configuring the envelope parameters.

The analyzer can also display input, output and sidechain waveforms, which are disabled by default. You can enable them in the time-graph settings (a button on the left).

Custom shape

Custom shape button displays a custom shape editor, which lets you control the output level of each event. The graph controls the mapping of input event level into its output level. Normally it is generated automatically from velocity markers in the analyzer, but you may want to specify it manually for special effects. The X axis is the input level that the detector computes for each input event (red dots). The Y axis is the desired output level (black dots). So for example a straight line from left bottom [-80dB, -80dB] to right top [0dB, 0dB] means no processing at all, because whatever the input event level is, the desired output level is the same. Please note that **Depth** parameter controls the amount of this processing, the same way as when using the velocity markers normally.

Plus

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

Minus

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

Rewind

Rewind 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).

Menu

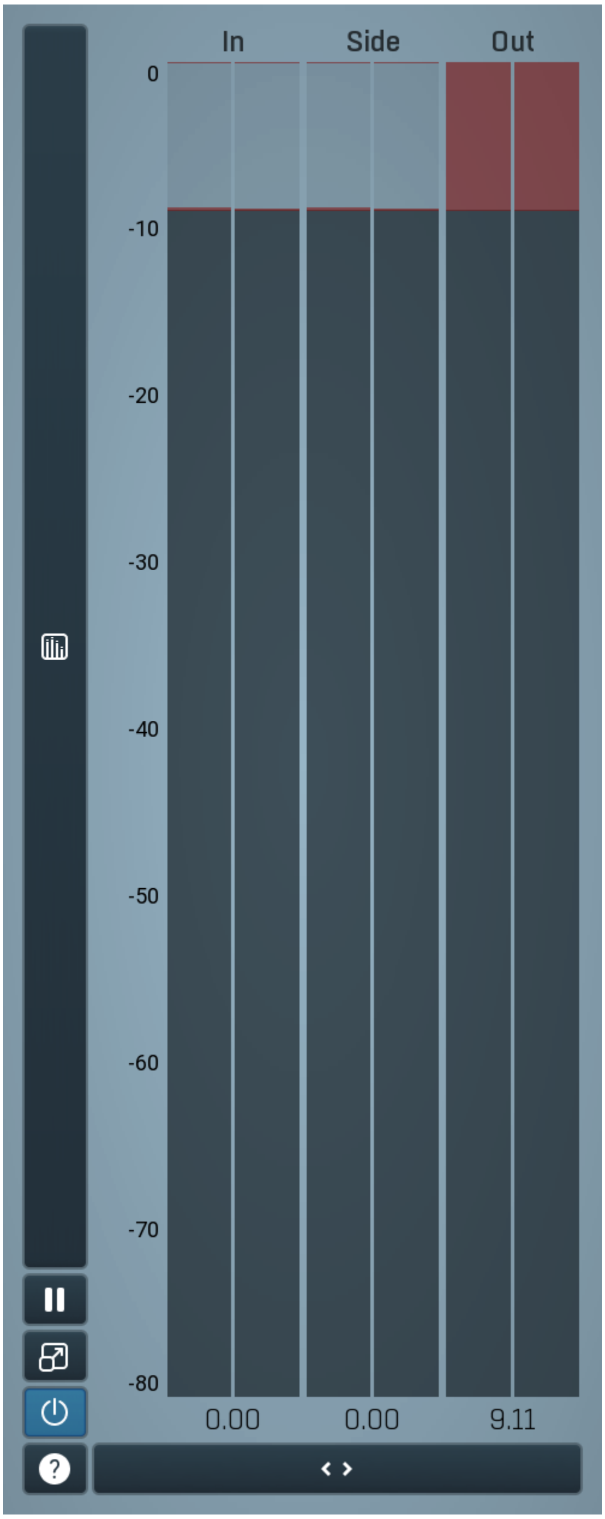
Menu button displays the time-graph settings. In this window you can control which graphs are displayed, the speed and other relevant parameters.

Pause

Pause button pauses the processing.

Enable

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



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 & Utilities** 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.

In meter indicates the total input level. The input meter shows the audio level before any specific processing (except potential oversampling and other pre-processing). It is always recommended to keep the input level under 0dB. You may need to adjust the previous processing plugins, track levels or gain stages to ensure that it is achieved.

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

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.

Time graph

Time graph 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.

Pause

Pause button pauses the processing.

Popup

Popup 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.

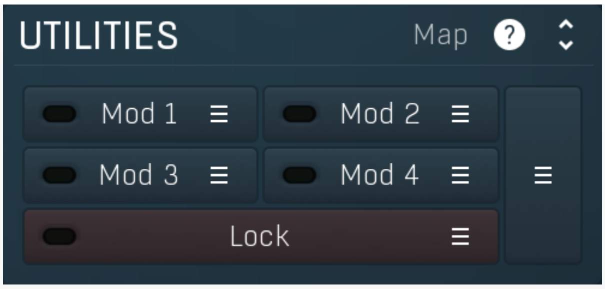
Enable

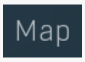
Enable button enables or disables the metering system. You can disable it to save system resources.

Collapse

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

Utilities





Map

Map button displays all current mappings of modulators, multiparameters and MIDI (whichever subsystems the plugin provides).

Modulator

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.

Menu

Menu 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.

Menu

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

Lock

Lock button displays the settings of the global parameter lock. Click on it using your left mouse button to open the Global Parameter Lock window, listing all those parameters that are currently able to be 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 currently-lockable parameters and then activates the learning mode. After that, every parameter you touch will be added to the lock. Learning mode is ended by clicking the button again.
The On/Off button built into the Lock button enables or disables the active locks.

Collapse

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

Multiparameter

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.

Menu

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

Learn attaches any parameters, including ranges. Click this, then move any parameters through the ranges that you want and click the multiparameter button again to finish. While learning is active, "REC" is displayed on the multiparameter button and learning mode is ended by clicking the button again.

Clear & Learn clears any parameters currently in the list then attaches any parameters, including ranges. Click this, then move any parameters through the ranges that you want and click the multiparameter button again to finish. While learning is active, "REC" is displayed on the multiparameter button and learning mode is ended by clicking the button again.

Reset resets all multiparameter settings to defaults.

Quick Learn clears any parameters currently in the list, attaches one parameter, including its range and assigns its name to the multiparameter. Click this, then move one parameter through the range that you want.

Attach MIDI Controller opens the MIDI Settings window, selects a unused parameter and activates MIDI learn. Click this then move the MIDI controller that you want to assign.

Reorder to ... lets you change the order of the multiparameters. This can be useful when creating active-presets. Please note that this feature can cause problems when one multiparameter controls other multiparameters, as these associations will not be preserved and they will need to be rebuilt.

In learning 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.

Collapse

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