MSpectralDynamics

Overview

MSpectralDynamics represents a true audio processing revolution. It's capable of a very wide range of effects, and includes top class features such as the 'custom processing shape'. It also contains a free-form linear-phase equalizer with a range from -80dB to 0dB, which allows you to fix specific problems in a recording. However its true strength is as a modern high end replacement for multiband compressors and loudness maximizers.

Multiband compressors have become a very popular tool for balancing the spectrum and maximizing loudness. However they are also very clumsy, their effects often sound unnatural and it's all too easy to destroy your audio material completely with them.

MSpectralDynamics uses a different approach as it works with the entire audio spectrum instead of bands. It approximates the energy located in each frequency and its surroundings and applies the dynamics to that frequency accordingly. This highly complex algorithm provides state-of-the-art sound quality and features that are simply impossible to achieve by any other method.

With spectral compression you can balance the spectral energy and increase loudness with few artefacts. You can also apply spectral expansion to excite frequencies and increase sound clarity. But this is just the beginning...

Introduction

Processor 1 (the panel on the left) is enabled by default and performs a little compression. On the Analyser view you can see the threshold of the processor displayed using a horizontal **red** line. The power spectrum of your recording is displayed using a **green line and area**. The threshold can be dragged with the mouse or adjusted in the Processor panel. When you move the threshold below the highest point of the spectrum, the green area splits from the green line. The green line defines the input signal (before any **Input gain** is applied) and the green area defines the output signal. This allows you to see the processor in action. A **blue line** now also appears on top of the analyser view and shows the gain reduction at each frequency across the spectrum.

As an example, let's say we want to maximize loudness, and balance frequencies in a recording. To do this we want to make the output spectrum (green area) as flat or horizontal as possible (with the exception of the low-end which usually needs to be cut-off to avoid problems with large monitors). Lower the threshold, increase the ration and the output gain. Using such a spectrum forces the recording to play all frequencies with a similar amount of power which our brain translates as louder, and therefore "nicer" sounding.

The lower the threshold, the more the spectrum is balanced, but also the more artifacts it can cause. In this case traditional distortion will rarely occur, however spectral artefacts may be the problem instead. These usually appear as "sounds from Mars", and are similar to those caused by mp3 compression with an insufficient bitrate for example.

Other parameters related to 'processor 1' can be found in the **Processor 1 panel**. Besides threshold, **ratio** is also very important. It generally defines how much of the power will be reduced (or increased) above the threshold. The goal is to find the optimum balance between the threshold and ratio to increase loudness as much as possible without causing artifacts.

Spectrum panel is also significant as it defines the way the spectrum is understood both by the plug-in and by you. You should keep a **High** quality setting whenever your CPU can handle it. Another primary parameter is **Smoothness** which generally defines how much each frequency affects surrounding ones. Setting higher values will yield more natural results but at the cost of reduced loudness, as even weak frequencies are affected by neighbouring ones.

Practical examples

We've demonstrated how spectral compression balances the spectrum and consequently increases loudness. However there are other, more advanced things that can be done with MSpectralDynamics. In fact we've barely explored the potential of this plugin.

Exciting dominant frequencies

By using MSpectralDynamics as an expander, already loud frequencies can be amplified further. Each processor can be switched to operate as a compressor or a downward expander (down arrow button is pushed). This changes the spectral envelope, on individual tracks it therefore changes the timbre of some instruments. If used on the master track it can bring the dominant frequencies to the front. Many mixes end up cluttered, because they are too crowded or over compressed. By using spectral expansion, you can restore the natural dynamics and let the listener focus on the dominant sounds in the mix without having to analyze the whole cluttered mix, where often the differences between dominant and subtle sounds are too low. This technique can also be used in sound design to create effects.

Denoising

A noise is by definition statistically unpredictable. Denoisers usually take the spectrum and decrease frequencies under a certain threshold as these frequencies are likely to be "just noise". This can be done with MSpectralDynamics, and with potentially better audio quality, as there are many options to adjust to the particular type of noise.
If the gate processor or processor 1 (in downwards expansion mode) is used, with a high threshold, then the level of many frequencies will be below it and the results may again resemble a badly encoded mp3 sound. However when you turn the threshold low enough, only silent frequencies and noise, will be processed. So how do we place the threshold just above the noise level? The easiest way is to find a part of the recording which only contains noise. Then use the Capture button to learn the noise profile and potentially adjust the threshold accordingly. Attack & Release parameters can then be used to make things sound as natural as possible. It is often wise to have a very low attack, although a higher release can be used, because any noise that appears shortly after a louder sound will likely be masked by it, making it disappear (known as the transient masking effect).

Avoiding collisions between tracks

Separating two instruments, which occupy the same space in the spectrum is a common mixing problem. A typical example is the frequency clashes of a bass guitar and bass drum. The usual approach is to use an equalizer on the colliding frequencies from one of the tracks. This is not an ideal solution however, as it permanently changes one of the tracks and removes the 'collision frequencies' even when the other instrument isn't actually playing and there is ample room in the spectrum. Another solution is to use a dynamic equalizer, MAutoDynamicEq or MDynamicEq for example, which will only remove these collision frequencies when necessary. However we can tackle this by demonstrating yet another way of using MSpectralDynamics.
As you might expect, we will be using the side-chain feature of MSpectralDynamics. We will use the above example of clashing frequencies between bass guitar and bass drum, but this method can be used with anything. Firstly we need to decide which track is more important. We will choose the bass drum track, so will be removing frequencies from the bass guitar track. Put MSpectralDynamics on the bass guitar track and send the bass drum to it as a side-chain (check your host sequencer for specific instructions on how to do this as methods vary). Activate the side-chain input using the Side-chain button in the Spectrum panel of MSpectralDynamics. The plugin now modifies the bass guitar, but measures frequencies from the bass drum. We want to reduce the bass frequencies, so the only thing we need to do is to set it up as a normal compressor. That's all! The great thing about this concept is that you don't need to worry about what frequencies are colliding, the plugin does all that for you!

Resynthesis from a noise

A slightly more exotic effect can be created using the side-chain and gate (or downwards expander). With this method you have two sounds, let's adopt the vocoder terminology and call the side-chain a modulator, while the main input will be the carrier. Both inputs can receive any signals, but for this example we will send white noise to the carrier.

To do this, we put MNoiseGenerator on an empty audio track, insert MSpectralDynamics, enable the gate and side-chain and send any other track to the side-chain (the modulator). The plugin will start removing those frequencies that do not exist in the modulator from the white noise, which theoretically contains all frequencies and therefore the filtered white noise starts resembling the modulator track. How closely the resynthesized sound follows the modulated audio depends on the settings used. Just remember that the point is to use this as a creative effect.

Easy screen vs. Edit screen

The plugin provides 2 user interfaces - an **easy screen** and an **edit screen**. Use the Edit button to switch between the two.

By default most plugins open on the **easy screen** (edit button released). This screen is a simplified view of the plugin which provides just a few controls. On the left hand side of the plugin you can see the list of available **devices / instruments** (previously called 'active presets'), that is, presets with controls. These controls are actually nothing more than multiparameters (single knobs that can control one or more of the plug-in's parameters and sometimes known as Macro controls in other plug-ins) and are described in more detail later. Each device may provide different controls and usually is intended for a specific purpose. The easy screen is designed for you to be able to perform common tasks, quickly and easily, without the need to use the advanced settings (that is, those available on the Edit screen).

In most cases the devices are highlighted using different text colors. In some cases the colors only mark different types of processing, but in most cases the general rule is that **black/white devices** are the essential ones designed for general use. **Green devices** are designed for a specific task or audio materials, e.g. de-essing or processing vocals in a compressor plugin. **Red devices** usually provide some very special processing or some extreme or creative settings. In a distortion plugin, for example, these may produce an extremely distorted output. **Blue devices** require an additional input, a side-chain or MIDI input usually. Without these additional inputs these **Blue** presets usually do not function as intended. Please check your host's documentation about routing side-chain and MIDI into an effect plugin.

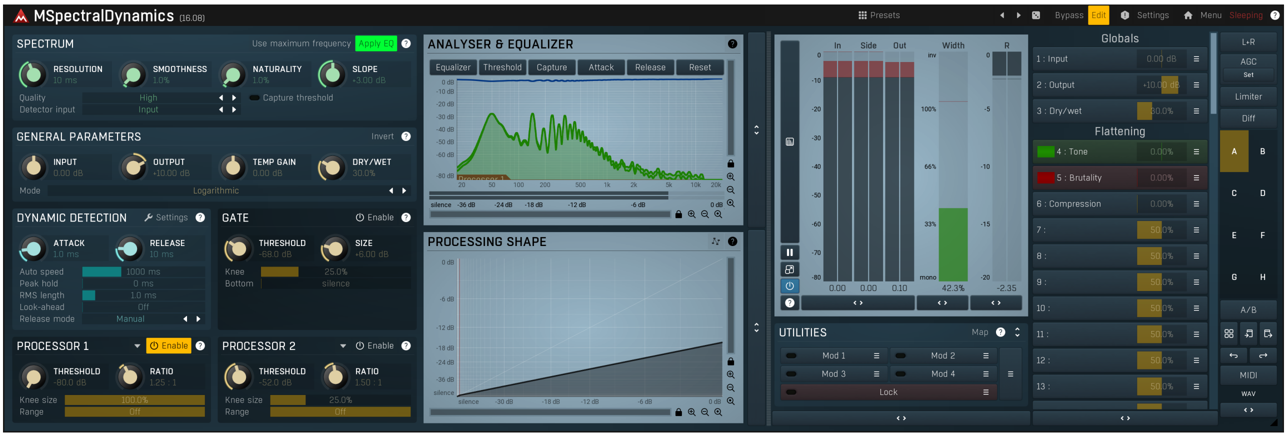
To the right of the controls are the meters or time-graphs for the plugin; the standard plugin Toolbar may be to the right of these or at the bottom of the plugin.

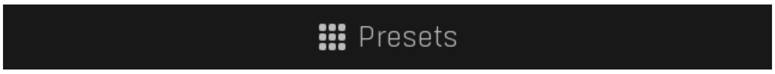
By clicking the **Edit button** you can switch the plugin to **edit mode** (edit button pushed). This mode provides all the of the features that the plugin offers. You lose no settings by toggling between edit mode and the easy screen unless you actually change something. This way you can easily check what is "under the hood" for each device, or start with an device and then tweak the plugin settings further.

Devices are factory specified and cannot be modified directly by users, however you can still make your own and store them as normal presets. To do so, configure the plugin as desired, then define each multiparameter and specify its name in its settings. You can then switch to the easy screen and check the user interface that you have created. Once you are satisfied with it, save it as a normal preset while you are on the easy screen. Although your preset will not be displayed or selected in the list of available devices, the functionality will be exactly the same. For more information about multiparameters and devices please check the **online video tutorials**.

If you are an advanced designer, you can also view both the easy and edit screens at the same time. To do that, hold **Ctrl** key and press the Edit button.

Edit mode





Presets

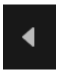
Presets button shows a window with all available presets. A preset can be loaded from the preset window by double-clicking on it, selecting via the buttons or by using your keyboard. You can also manage the directory structure, store new presets, replace existing ones etc. Presets are global, so a preset 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

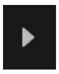
Files are named based on the name of the plugin like this: "{pluginname}.presets", so for example MAutopan.presets or MDynamics.presets. If the directory cannot be found on your computer for some reason, you can just search for the particular file.

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

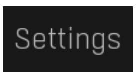
Right arrow button loads the next preset.

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 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.

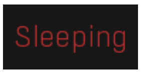
Global system settings contains some settings for all MeldaProduction plugins. Once you change any of them, restart your DAW if needed, and it will affect all MeldaProduction plugins.

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

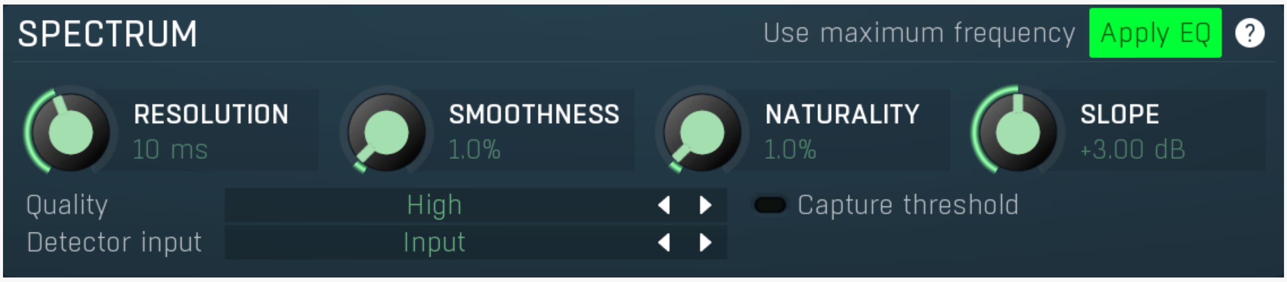
WWW button shows a menu with additional information about the plugin. You can check for updates, get easy access to support, MeldaProduction web page, video tutorials, Facebook/Twitter/YouTube channels and more.



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.

Spectrum panel



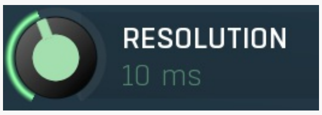
Spectrum panel contains properties that affect how the spectral information is detected and processed. Please note that all of these parameters affect the processing too, which is not true for most MeldaProduction plug-ins containing a spectral analyser.

Use maximum frequency

Use maximum frequency makes the processor use maximum magnitude of all the frequencies and apply them to the whole spectrum. It is useful for example to compress spectrum where the problematic frequency is changing.

Apply EQ

Apply EQ makes the processor use the EQ curve on the audio signal. You can disable it, if you want to change the detector signal only.



Resolution

Resolution defines how accurately the analyser can control attack and other parameters. A lower value causes the processing to be more accurate but require much more CPU power. This parameter does not have significant impact on the quality of resulting signal, however it controls how precisely the requested time domain parameters are changed, mostly **Attack**. Optimally the value should be lower than the attack value.

Range: 1.0 ms to 50 ms, default 10 ms

Smoothness

Smoothness makes the analyser smooth out the curve, so it contains less bumping up and down. It approximates the energy in each frequency and the resulting graph may be easier to understand. Higher smoothness value typically produces more natural results, since it removes unwanted peaks and troughs.

Range: 0.00% to 20.0%, default 5.0%

Naturality

Naturality makes the processor smooth out the processing curve, so that it avoids disruption of frequency correlations, that may cause some artefacts. A higher value produces more natural results, however requires more CPU power and lowers the processing precision.
Range: 0.00% to 20.0%, default 5.0%

Slope

Slope makes the analyzer increase magnitude of higher frequencies, since they are typically lower in energy. 3dB per octave is a default value, which makes the graph horizontal if it contains equal energy in each octave. It helps you understand the spectral properties of your audio signal and can be also used to change the global output signal character like this: a lower value typically creates a descent, which makes higher frequencies fall below the threshold so only the low-end is processed. Conversely a higher value lets the higher frequencies exceed a threshold and get processed more than the low-end.
Range: 0.00 dB to +6.00 dB, default +3.00 dB

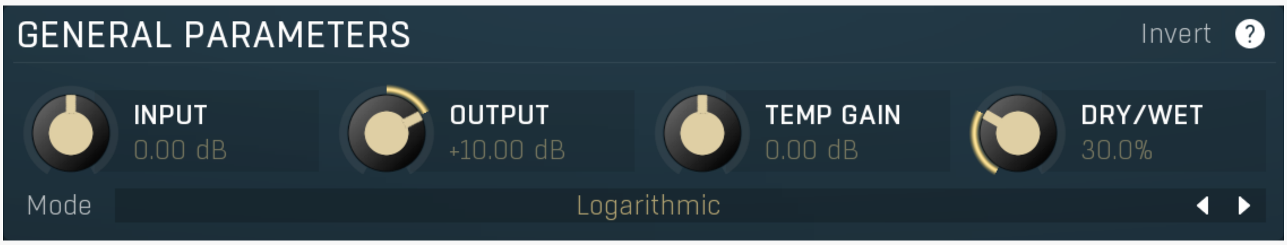
Quality

Quality defines the analyser processing quality, therefore the quality of the output signal. It can have significant impact on the performance. Please note that different quality settings also produce different sound character, so you should not find optimal settings in low quality mode and then switch to the high quality mode to render the project, since the sound would be simply different.

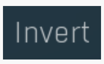
Detector input

Detector input controls which input buses are analyzed and how.
Input mode is default and makes the plugin analyse the main plugin input only.
SideChain activates the analysis of the side-chain input only and apply the processing to the main input. This can be used for ducking one main input based on the side-chain key input for example.
Input + SideChain analyzes both inputs and sums the levels of individual frequencies in both. This can provide one way to automatically but subtly avoid collisions. By applying it on one of the colliding tracks using a compressor, you can compress the input track only if both tracks exceed a threshold when summed.
Input - SideChain subtracts the levels of individual frequencies in the side-chain from the main input. When used as a compressor, it lets you exclude frequencies present in the side-chain from being compressed. When used as a gate it can do the opposite - by reducing the level of these frequencies it prevents them from exceed the threshold, so it can be seen as one way to solve collisions.
SideChain - Input works the other way around. It can be used to attenuate frequencies present in the side-chain, but not those present in the main input, hence cleaning the input signal.
Multiply multiplies the levels of individual frequencies. This may be the best way to solve collisions, since this operation produces frequencies present in both inputs at the same time.
Maximum produces maximum and **Minimum** produces minimum from individual frequencies in both inputs. Minimum can work similarly to Multiply mode and Maximum can be similar to Plus, but you could say that the strength of the effect doesn't add with both inputs as it can never rise above (maximum) or lower below (minimum) the levels in each input.

General parameters panel

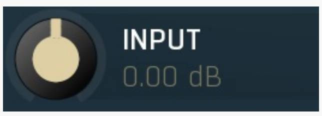


General parameters panel contains the parameters related to dynamic processing.



Invert

Invert reverts the polarity of the compressed signal, so the **Dry/Wet** can then be used to invert the transfer shape. It can be used to convert a gate into a ducker for example, dry/wet can then serve as amount of ducking.

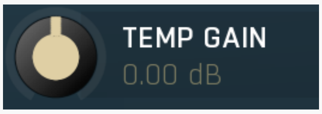


Input gain

Input gain defines gain applied to the incoming signal. If you set ratio to 1:1 and custom shape is disabled, then the plug-in works simply as a fast gain processor.
Range: -24.00 dB to +24.00 dB, default 0.00 dB

Output gain

Output gain defines the gain applied to the output signal. If you set ratio to 1:1 and custom shape is disabled, then the plug-in works simply as a fast gain processor.
Range: -24.00 dB to +24.00 dB, default 0.00 dB



Temp gain

Temp gain defines the temporary gain applied to the input signal and then reversed on the output. You can achieve the same effect by setting **Input gain** to a value **G** and **Output gain** to value **-G**. Moreover, this plug-in tries to approximate the gain reduction. Absolutely accurate approximation is not possible; however when you set the parameters so that the level is touching the threshold with temporary gain at 0dB, then any change to the temporary gain should change the amount of compression but keep the output level stable. Therefore the temporary gain in fact controls **amount of compression**.
Range: -24.00 dB to +24.00 dB, default 0.00 dB

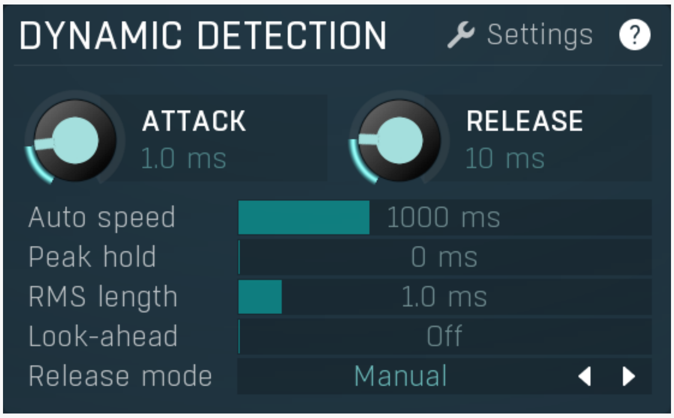
Dry/Wet

Dry/Wet defines the ratio between dry and wet signals. 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 the dry/wet ratio control to reduce 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 the ratio does NOT have the same effect as lowering dry/wet in most cases.
Range: 0.00% to 100.0%, default 100.0%

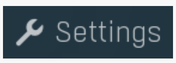
Mode

Mode affects the processing shape. The plug-in features special non-linear transfer shapes which affect the way the signal is processed. **Logarithmic** produces classic dynamic processing where a signal exceeding the threshold by 10dB at a compression ratio of 2 : 1 produces 5dB attenuation in output level. In this same scenario, **Squared** mode produces a slightly greater output attenuation of 6.4dB and **Linear** mode produces a still greater value of 7.5dB. Thus, Squared and Linear modes produce progressively more compression / expansion. There is no compromise in sound quality between the different modes. Comparing the three modes, Linear mode requires the least amount of CPU power, and Logarithmic the most.

Dynamic detection panel



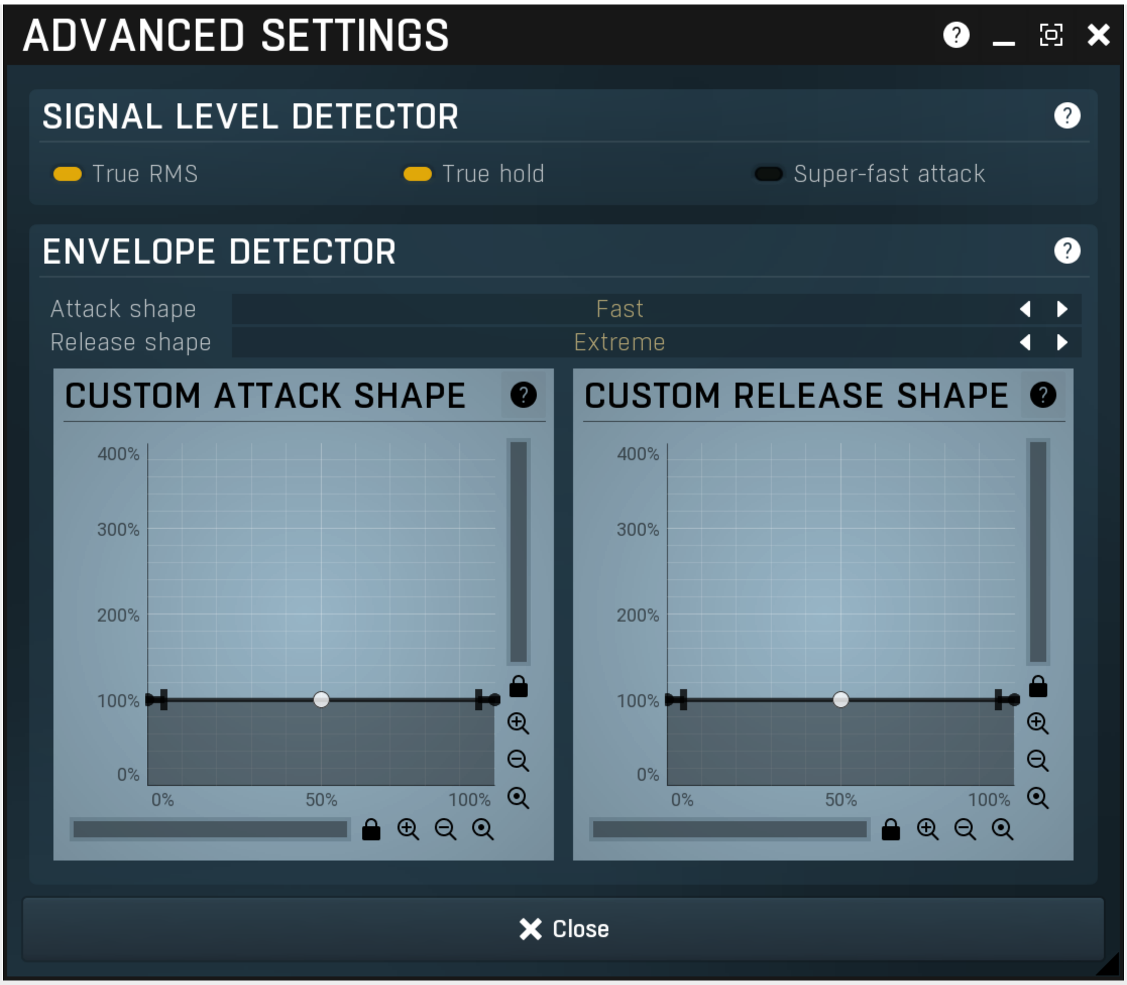
Dynamic detection panel contains the parameters defining how the plug-in determines the level of the source signal.



Settings

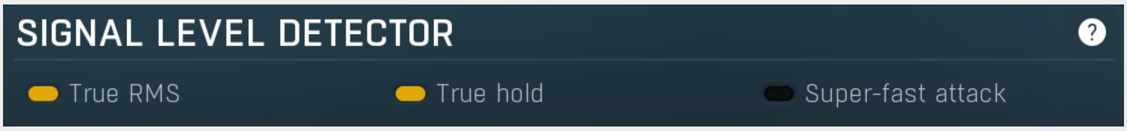
Settings button shows additional dynamics detector settings.

Advanced settings



Advanced settings contains more esoteric and advanced settings of the level detector. These include various kinds of detector signal preprocessing, attack & release responses and custom shapes, etc.

Signal level detector



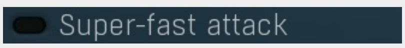
True RMS

True RMS enables the true RMS calculation instead of the simplified approximation with a slightly different response. When disabled, the calculation is faster and requires almost no memory, however it is also inaccurate. This may not necessarily be a disadvantage, but it may be worth checking the true RMS processor, which provides the standard RMS calculation with the response you would expect. True RMS processing is not much slower than the approximated version, but requires a considerable amount of memory.

True hold

True hold enables the true peak hold algorithm. When disabled, hold is implemented using a special filter which catches peaks and maximizes the level detector signal input by those peaks. In time the peaks decrease in level according to the **hold** parameter. This is effective, requires almost no CPU and memory is required, but it is also inaccurate. For example, since the peaks are not keeping their levels, it cannot be used along with the **look-ahead** feature to avoid distortion in limiters.

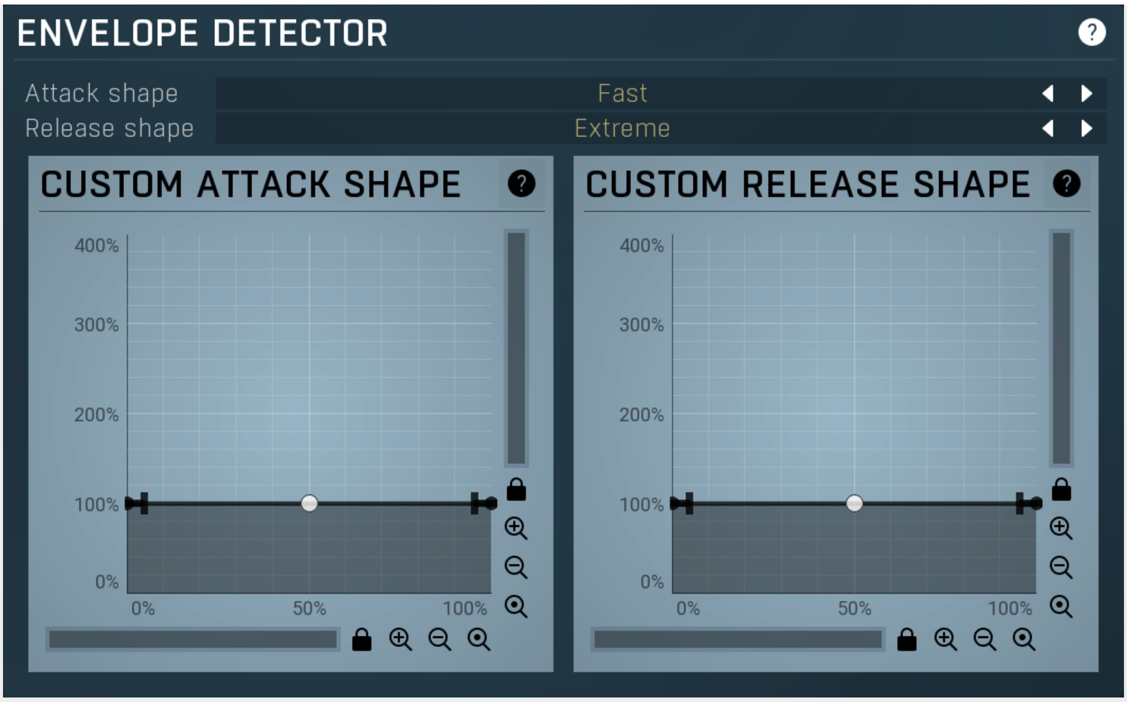
True hold, on the other hand, implements the fastest currently-known algorithm to provide the true peak hold response; this does not decay in time and correctly tracks peaks. The typical use in limiters, for example, is to use the same **hold** and **look-ahead**values - the **look-ahead** gives the limiter time and **hold** tracks the highest peaks ahead of the actual dynamic processing. This can highly improve the audio quality by removing unwanted distortion.



Super-fast attack

Super-fast attack ensures the level will never go below the threshold, allowing the dynamic processor to react as quickly as possible, even if **attack** time is higher than 0ms. This is specifically designed for compression and is incompatible with gating and any downwards processing. Note that if you use a soft knee, you may expect gain reduction even if the audio level is very low, or even silence for that matter.

Envelope detector



Attack shape

Attack shape controls the shape of the attack stage. The shape mainly affects the **ratio between pumping and distortion**, which simply cannot be avoided. Please note that the attack time parameter is quite dependent upon the mode, so you may expect differences in the actual attack time for different modes of the Attack shape.

Slow modes usually produce more pumping, but less distortion, as the detected level follows the input level more slowly. Conversely **Fast modes** reduce pumping, but cause more distortion. The type of the distortion is different between modes. You may actually profit from the distortion caused by some modes as the generated higher harmonics may enhance the audio. The default **Fast mode** provides a good compromise between distortion and pumping.

There are also **2 custom modes** available. With these modes you can actually draw the shape. Note that what you draw is NOT what you get. The custom shape graph converts the difference between the input level and the current detected level (as represented by the X-axis) into the speed of level detection (as represented by the Y-axis).

*For example, if you set the graph to show 100% across the X axis, then the results will be similar to the **Slow mode**. As the graph is flat, the speed of the detector is the same for all differences between the input and detected levels. If you then move the point on the right upwards to say 400%, it will mean that, if there is a big difference in the levels (a high X value), the detected level will follow the input level 400% faster than it normally would. The closer the detected level gets to the current audio level (a lower X value), the slower the change in the detected level. Similarly, if you take the point on the left and move it downwards to 0%, it will slow down the change to the detected level as it approaches the audio level (a low X value).*

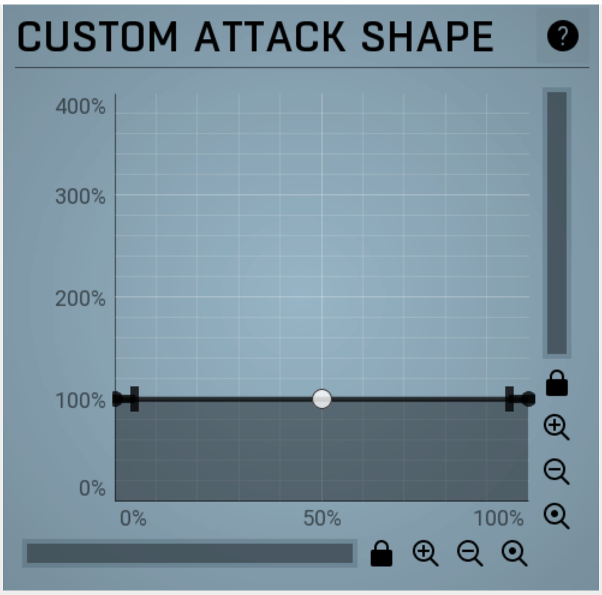
Release shape

Release shape controls the shape of the release stage. The shape affects the **ratio between pumping and distortion**, which simply cannot be avoided. Please note that the release time parameter is quite dependent on the mode, so you may expect differences in actual release time for different modes of the Release shape.

Slow modes usually producemore pumping, but less distortion, as the detected level follows the input level more slowly. Conversely **Fast modes** reduce pumping, but cause more distortion. The type of the distortion is different between modes. You may actually profit from the distortion caused by some modes as the generated higher harmonics may enhance the audio. The default **Fast mode** provides a good compromise between distortion and pumping.

There are also **2 custom modes** available. With these modes you can actually draw the shape. Please note that what you draw is NOT what you get. The custom shape graph converts the difference between the input level and the current detected level (as represented by the X-axis) into the speed of level detection (as represented by the Y-axis).

*For example, if you set the graph to show 100% across the X axis, then the results will be similar to the **Slow mode**. As the graph is flat, the speed of the detector is the same for all differences between the input and detected levels. If you then move the point on the right upwards to say 400%, it will mean that, if there is a big difference in the levels (a high X value), the detected level will follow the input level 400% faster than it normally would. The closer the detected level gets to the current audio level (a lower X value), the slower the change in the detected level. Similarly, if you take the point on the left and move it downwards to 0%, it will slow down the change to the detected level as it approaches the audio level (a low X value).*

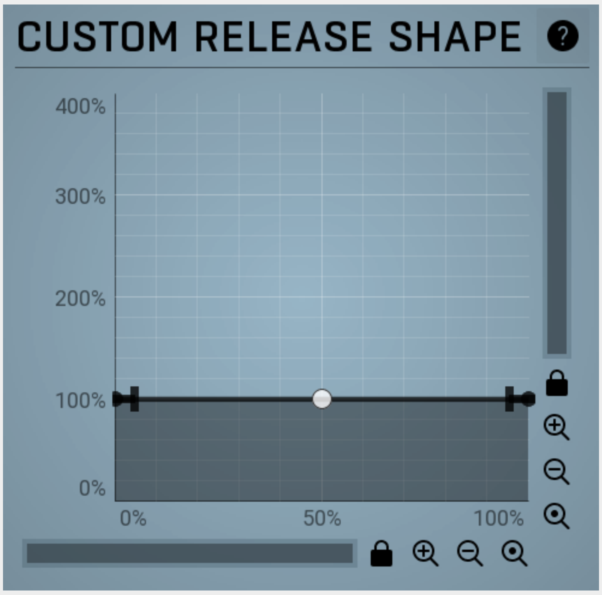


Custom attack shape

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.



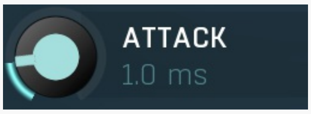
Custom release shape

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.



Attack

Attack defines the attack time, that is how quickly the level detector increases the measured input level. 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 measured 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.

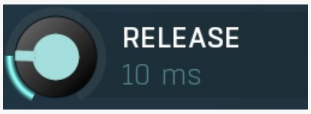
To understand the working of a level detector, it is best to cover the typical cases:

*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 5000 ms, default 100 ms

Auto speed

Auto speed defines how quickly the automatic release works. Specifically how much the release time increases/decreases per second. It is relevant only in **automatic** release modes. *For example if you set it to 5000ms, the release time will be able to increase by 1000ms in 5000ms, when incoming signal exceeds the lowest threshold.*
Range: 1.0 ms to 10000 ms, default 1000 ms

Peak hold

Peak hold defines the time that signal level detector holds its maximum before the release stage is allowed to start. As an example, you can imagine that when an attack stage ends there can be an additional peak hold stage and the level is not yet falling, before the release stage starts. This is true only when **true peak** mode is enabled (check the advanced detector settings if available).

It is often used in **gates** to avoid the gated level falling below the threshold too quickly, while having short release times. If you want the gate to close quickly, you need a short release time. But in that case the ending may be too abrupt and even cause some distortion. So you use the peak hold to delay the release stage.

It is also used along with **look-ahead** to avoid distortion in **limiters and compressors**. If you need a very short attack, the attack stage may be too quick and cause distortions. In limiters this attack time is often 0ms, in which case it becomes a clipper. Setting look-ahead and peak hold to the same value will make the detector move ahead in time, so that it can react to attack stages before they actually occur and yet hold the levels for the actual signal to come.
Range: 0 ms to 1000 ms, default 0 ms

RMS length

RMS length smoothes out the values of the input levels (not the input itself), such that the level detector receives the pre-processed signal without so many fluctuations. When set to its minimum value the detector becomes a so-called "peak detector", otherwise it is an "RMS detector".

When you look at a typical waveform in any editor, you can see that the signal is constantly changing and contains various transient bursts and separate peaks. This is especially noticeable with rhythmical signals, such as drums. Trying to imagine how a typical attack/release detector works with such a wild signal may be complex, at least. RMS essentially takes the surrounding samples and averages them. The result is a much smoother signal with fewer individual peaks and short noise bursts.

RMS length controls how many samples are taken to calculate the average. It stabilizes the levels, but it also causes a slower response time. As such it is great for mastering, when you want to lower the dynamic range in a very subtle way without any instabilities. However, it is not really desirable for processing drums, for example, where the transient bursts may actually be individual drum hits, hence it is usually recommended to use peak detectors for percussive instruments.

Note that the RMS detector has 2 modes - a simplified approximation is used by default, and a true RMS is processor can be enabled from the advanced settings (if provided). Both respond differently, neither of them is better than the other, they are simply different.
Range: Peak to 100 ms, default 1.0 ms

Look-ahead

Look-ahead delays the actual signal being processed, but keeps the detector signal intact. This makes the processor use a signal that has not actually arrived for dynamic calculation. This allows the processor to respond even faster, in fact, ahead of time. This feature is useful for mastering, however it naturally induces latency.
Look-ahead can be available in milliseconds (with obvious meaning) or in percentages. In percentages the look-ahead delay is computed automatically based on the attack and hold times. For example, if look-ahead is 100%, attack time 2ms and peak hold 10ms, then the look-ahead is 10ms; 60% look-ahead would be 7.2ms. If the look-ahead is simply an on/off switch, then it is toggling between 0% and 100% values.

Before using look-ahead, you should understand what such a feature does exactly as the results can potentially be damaging to your audio. Look-ahead basically moves the signal back in time, in other words its signal detector measures the input levels ahead of time. This means that when the detector is in the attack stage, the level is rising, the actual signal is not rising yet, but it will do so soon. However, the same applies to the release stage! When the detector moves to the release stage, the actual signal is not falling yet. This can lead to very strange artifacts (which can be used creatively of course).

The common way to fix this is to set the **release time** considerably higher than the **attack time**. In this way, the level will rise ahead of time in the attack stage, and same will happen for the release stage and the level will go down, however, since the level is falling slowly, the look-ahead will not be that relevant.

Another option is to use the **peak hold** feature. It is highly recommended to enable **true hold** in the advanced detector settings if available. Essentially this feature maximizes the input level over a certain period of time. *So for example, if you set look-ahead to 5ms and peak hold to 5ms as well, the actual signal will arrive 5ms later than the detector signal, however the peak hold feature will ensure that the detector holds the highest peaks for 5ms, so the attack stage will be ahead of time, but the release will not! You can consider it a form of latency compensation for the release stage.*

*Look-ahead is commonly used in **limiters** along with very low (often 0ms) attack times to avoid distortion. With 0ms attack time the limiter is immediately following the input and when the level gets above 0dB, it turns it down to 0dB, so the attack stage is effectively being clipped. To avoid distortion produced by this effect, you can increase look-ahead and peak hold to the same value, say 1ms. As a*

result the attack stage occurs before it actually occurs, so the distortion is still present, but in much lower levels and usually is masked by the forthcoming transient.
Range: Off to 1000 ms, default Off

Release mode

Release mode defines how the plug-in performs when decreasing level. In **manual mode** this is based only on the **release time**, which is suitable for most cases when the signal has constant characteristics. Automatic release modes can adapt to signals with unstable characteristics.

Automatic and **Automatic fast** modes: the longer the level stays above the threshold, the longer the release time will be and thus, the longer it will take to move below the threshold and end the release stage. The idea is that if the input is loud for some time, it will most likely stay that way for some more time, hence it should be stabilized to avoid unnecessary temporary fluctuations, which could result in pumping.
Both automatic modes increase the release time when the input signal is above the threshold and vice versa. The speed of the increase depends on the **Auto speed** parameter. Automatic fast mode uses full speed immediately after crossing the threshold, automatic mode varies the speed according to the current signal level.

For example, when a guitarist plays softly, the level is low and fluctuates around the threshold and the release time gets slower. So the processor quickly responds to sudden changes. However, when the guitarist starts playing a solo, the level rises and, the longer the solo is, the longer the release time becomes, hence the response becomes slower avoiding unnecessary fluctuations (pumping) when the solo contains small silent sections.

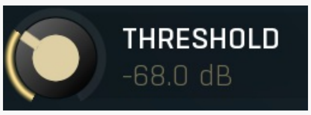
Linear 1 and **Linear 2** modes: the higher the level is, the longer the release. The idea is that if the input is very loud, it will probably stay that way for some time, so it is wise to keep the levels up too. This is similar to the automatic modes, however the main factor is not how long the level is high, but how high it is.
Below the threshold the release time is the same as the attack time, above the threshold the release time rises from the attack time up to the specified release time parameter. Linear 1 mode usually provides higher release times than does Linear 2.

Opto mode: the higher the level is, the shorter the release. So this is kind of the opposite of linear modes. The idea is, that you are expecting short transients, which you wish to deal with. Normally the higher the level would get in such a transient, the longer it would take to get the level below the threshold, so, when used in a compressor for example, these transients would cause unnecessary compression in the sustain stage. The opto detector lowers the level quickly, minimizing the amount of compression in the sustain stage.

For example, let's say you are compressing a full drumset, but there is a very dominant sharp and short hi-hat sound, so it is appropriate to have short release times. You would use ***Opto*** *mode. But the rest of the drumset deserves a softer treatment, so you want to keep longer release times. Use one of the other modes.*

Gate panel

Gate panel contains parameters for the noise-gate.



Threshold

Threshold determines the maximum signal level below which the effect starts to apply.
Range: -80.0 dB to 0.00 dB, default -68.0 dB

Size

Size defines size of the interval between the gate threshold and point when the output signal level reaches zero.
Range: 0.00 dB to +24.00 dB, default +6.00 dB

Knee

Knee defines the size of the smoothing knee.
Range: 0.00% to 100.0%, default 25.0%

Bottom

Bottom defines the volume reached when the gate is fully closed, hence the Threshold minus Size. In most cases you can leave this set to **silence**.
Range: silence to -80.0 dB, default silence

Processor 1 panel

Processor 1 panel contains parameters of the primary processor, which can behave like a compressor or expander.

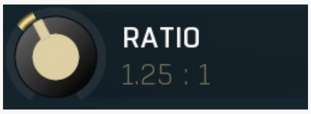
Downwards

Downwards button switches the processor into a downward. expander. In this mode the processor reacts to levels below the threshold, instead of above the threshold as in normal mode. Despite this, upwards compression can be done too. This mode is particularly useful for expansion, since upward expansion is somewhat dangerous as it can significantly amplify the audio way above 0dB.

Compression reduces the dynamic range of sounds above the threshold level, reducing the level over the threshold by the ratio. For a ratio of 1.50:1, 9 dB over the threshold will be reduced to 6 dB over. Levels below the threshold are not changed.
Downward expansion increases the dynamic range of sounds below the threshold level, reducing the level under the threshold by the ratio. For a ratio of 1.50:1, 9 dB under the threshold will be reduced to 13.5 dB under. Levels above the threshold are not changed.

Threshold

Threshold determines the minimum signal level above which the compression effect starts to apply.
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. The higher the ratio, the more compression you get.
Range: 1 : 1.50 to Infinity, default 1.50 : 1

Knee size

Knee size defines size of the knee.
Range: 0.00% to 100.0%, default 25.0%

Range

Range defines size of the interval above the threshold after which the original signal ratio is restored.
Range: +1.00 dB to Off, default Off

Processor 2 panel

Processor 2 panel contains parameters of the secondary processor, which can behave like a compressor or expander.

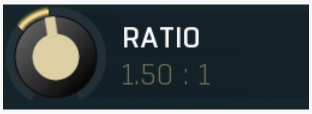
Downwards

Downwards button switches the processor into a downward. expander. In this mode the processor reacts to levels below the threshold, instead of above the threshold as in normal mode. Despite this, upwards compression can be done too. This mode is particularly useful for expansion, since upward expansion is somewhat dangerous as it can significantly amplify the audio way above 0dB.

Compression reduces the dynamic range of sounds above the threshold level, reducing the level over the threshold by the ratio. For a ratio of 1.50:1, 9 dB over the threshold will be reduced to 6 dB over. Levels below the threshold are not changed.
Downward expansion increases the dynamic range of sounds below the threshold level, reducing the level under the threshold by the ratio. For a ratio of 1.50:1, 9 dB under the threshold will be reduced to 13.5 dB under. Levels above the threshold are not changed.

Threshold

Threshold determines the minimum signal level above which the compression effect starts to apply.
Range: -80.0 dB to 0.00 dB, default -52.0 dB



Ratio

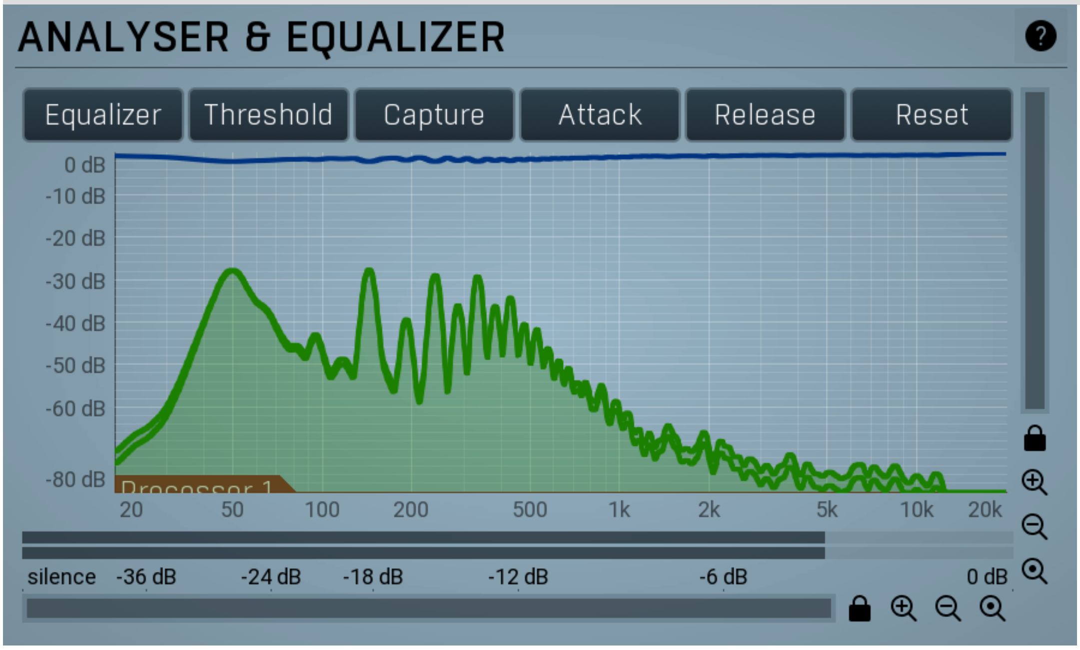
Ratio defines the compression ratio of the input signal above the threshold. The higher the ratio, the more compression you get.
Range: 1 : 1.50 to Infinity, default 1.50 : 1

Knee size

Knee size defines size of the knee.
Range: 0.00% to 100.0%, default 25.0%

Range

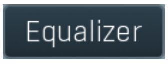
Range defines size of the interval above the threshold after which the original signal ratio is restored.
Range: +1.00 dB to Off, default Off



Analyzer & Equalizer panel

Analyzer & Equalizer panel contains the dynamic analysis of the signal and a free-form equalizer that is applied before the signal is processed through the dynamic processor. The equalizer has no CPU requirements, so do not be afraid to use it to correct the source material and remove unwanted frequencies etc.
On top you can see 2 buttons - Equalizer and Threshold. The first you already know about. The latter is attached to the **Processor 1**

Threshold, and it allows a free-form Threshold graph to be drawn. This way you can define the amount of processing for each frequency. The **green area** contains the signal power of the output signal after processing. The **green line** describes the signal power of the incoming signal before processing. The **blue line** on top shows the gain reduction for each frequency. The **horizontal orange lines** indicate the thresholds of enabled processors unless the custom shaping mode is enabled.



Equalizer
Equalizer button shows or hides the free-form equalizer editor.

Threshold
Threshold button shows or hides the free-form threshold editor.

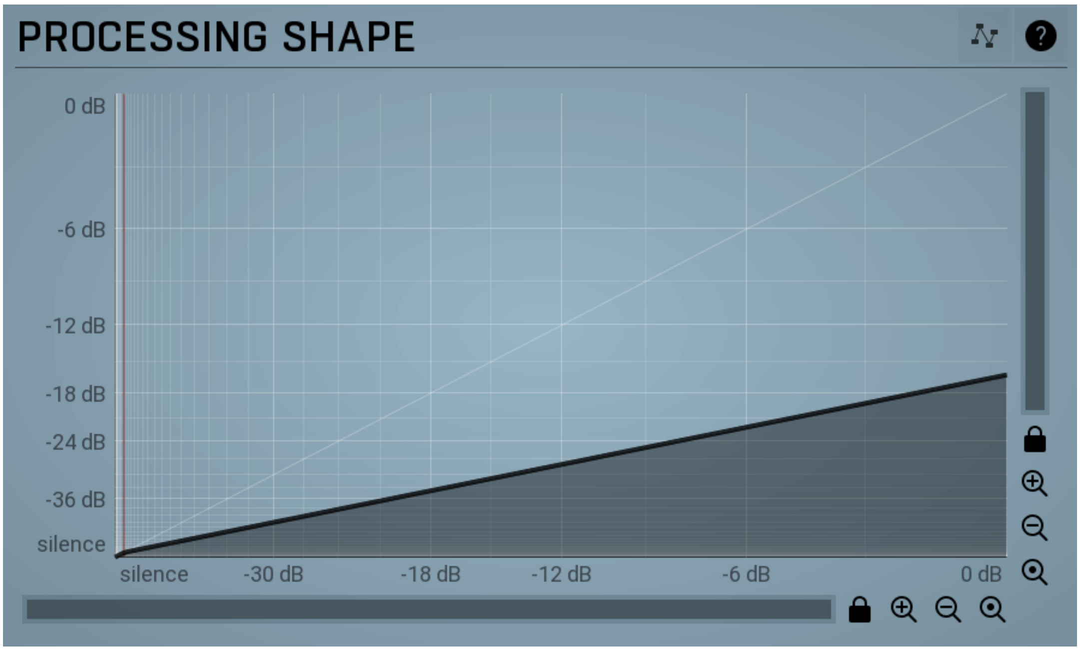
Capture
Capture button lets you analyze the input and set the threshold graph accordingly. This can be used to provide a very transparent spectral compression or denoising for example. Enable the playback, then click the button to initiate the analysis. Click the button again when finished.

Attack
Attack button shows or hides the free-form attack editor.

Release
Release button shows or hides the free-form release editor. Please note that this is functional only in **Manual** release mode.

Reset
Reset button restores the original settings of the current free-form graph. If you hold **Ctrl** while clicking then all 4 graphs are reset.

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



Level shape graph

Level shape graph displays the dynamic processing transformation shape. The X axis represents the input signal level, Y axis defines the output level.

Please note that this display is not logarithmic. This can lead to confusion, as, for example, a moving expander's threshold changes the graph's slope while the ratio stays the same. This is however necessary, because a logarithmic display can never contain silence, as it is minus infinity decibels, and the silence point is essential for gates for example. The display is therefore a compromise between usability and accuracy.

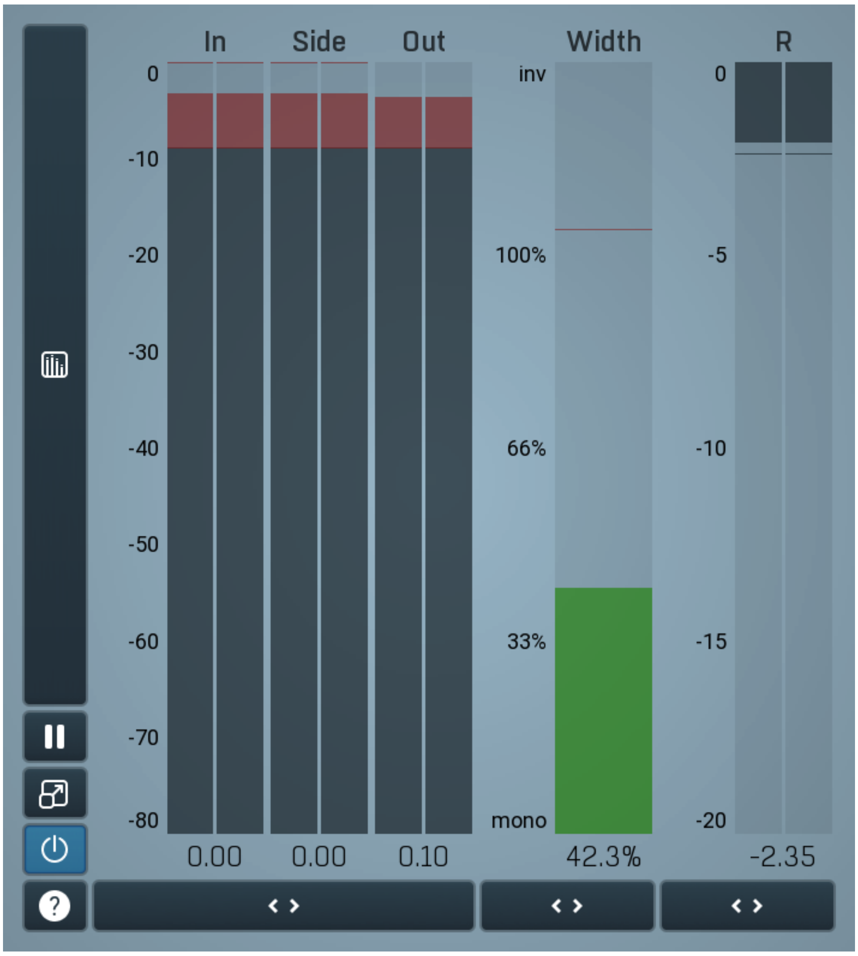
The moving vertical line shows the current detected level. It may be moving extremely quickly depending on the settings. It may also be invisible if the input level is silence or above 0dB (which is not recommended unless you are using the processor as a limiter). There may be other graphs available, such as input & output waveform and gain reduction time graphs.

Graph

Graph button enables or disables the custom level shape. When enabled, it inherits the automatic settings and you can draw any processing shape you want.

Collapse

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



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.

R meter shows gain reduction for each channel. Negative values, running down from the top, mean that compression or limiting is occurring. The lower the value, the stronger the effect. For maximum transparency you should try to achieve the least amount of gain reduction. Expansion is not indicated in this meter.

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.

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.

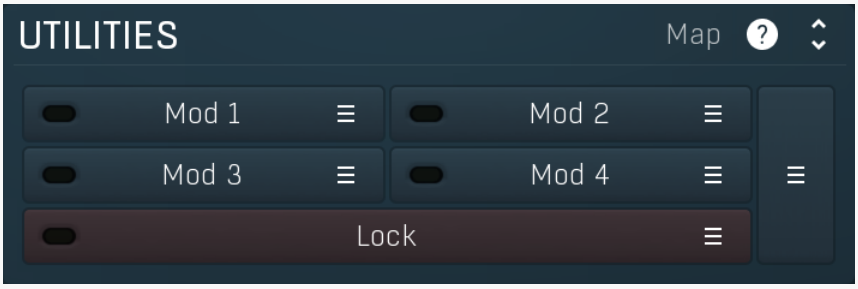
Enable
Enable button enables or disables the metering system. You can disable it to save system resources.

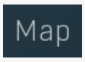
Collapse
Collapse button minimizes or enlarges the panel to release space for other editors.

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

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.