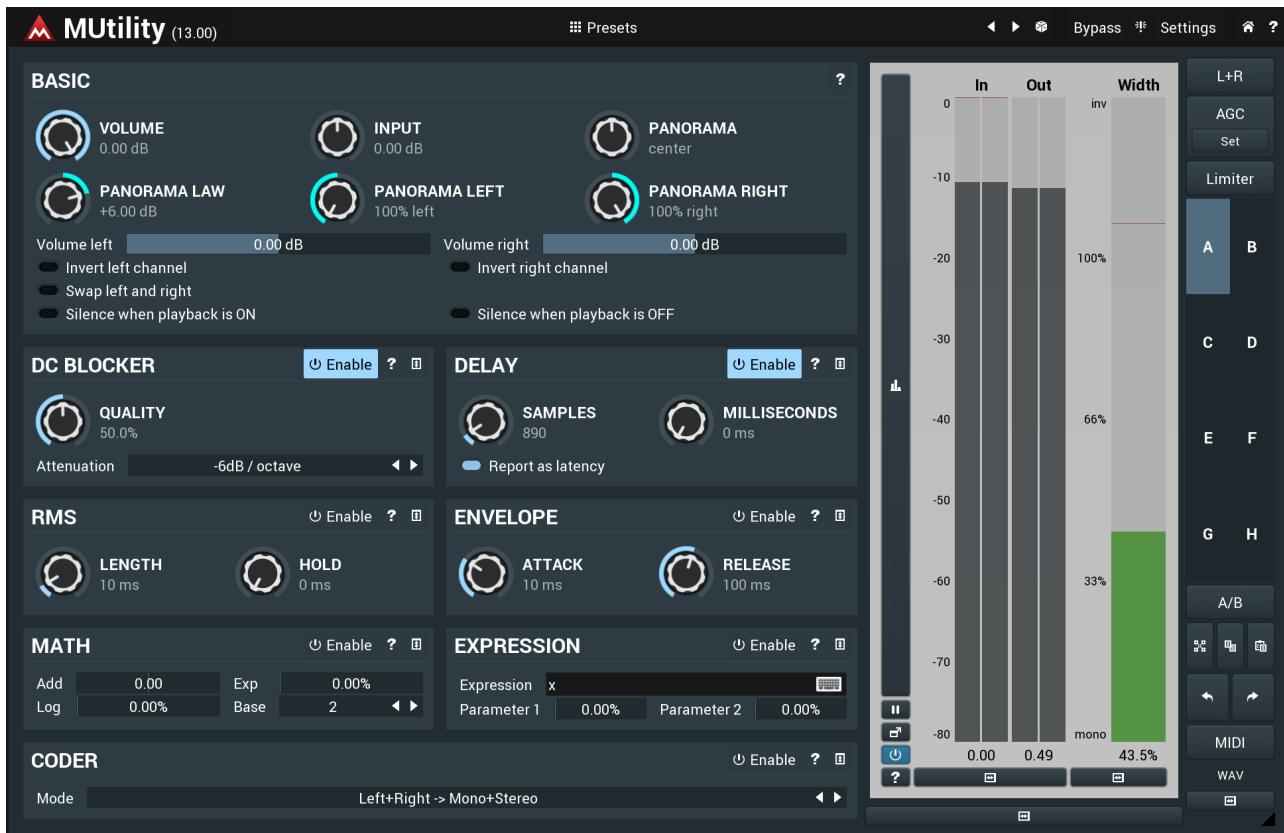


# MUtility



## Presets

## Presets

### button

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

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

Presets can be backed up by using either the Export button, or by saving the actual preset files, which are found in the following directories:  
Windows: C:\Users\{username}\AppData\Roaming\MeldaProduction  
Mac OS X: ~/Library/Application support/MeldaProduction

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

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



### Left arrow button

Left arrow button loads the previous preset.



### Right arrow button

Right arrow button loads the next preset.



### Randomize button

Randomize button loads a random preset.



### Panic button

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

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

**Activate** lets you activate the plugin if the drag & drop activation method does not work in your host. In this case either click this button and browse to the licence file on your computer and select it. Or open the licence file in any text editor, copy its contents to the system clipboard and click this button. The plugin will then perform the activation using the data in the clipboard, if possible.

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

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

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

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



### WWW button

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.

## Basic panel

Basic panel contains basic audio processing features.

### Volume

Volume defines the input volume adjustment, which is an alternative to gain with a different range.

Range: silence to 0.00 dB, default 0.00 dB

### Input gain

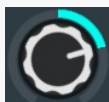
Input gain defines the input gain.

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

### Panorama

Panorama defines the panorama applied to the input.

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



**PANORAMA LAW**  
+6.00 dB

### Panorama law

Panorama law defines how the panner compensates for the loss of gain when the signal is panned. When the panorama is in the center, no change is performed whatsoever. However when the signal is panned fully to the left for example, right channel is silent and left contains mix of left and right input channels. The question is, what the "mix" is exactly.

Let's say the audio signal is monophonic. Then taking (left + right) as the new left signal is reasonable. It contains the signal from both input channels. However it also has higher level, +6dB if the input was monophonic. This means the panorama law of +6dB. If we choose (left + right) / 2 instead, panorama law of 0dB, then the output will have the same level, but the overall energy will be -6dB lower, because there were originally 2 channels, now there's just one as the right channel is silent.

There is no correct answer here. The basic suggestion is to choose panorama law of 0dB-3dB, if you don't want the peak level to rise. +6dB provides usually the natural results as it keeps the energy. A pan law of -3dB is often used for stereo tracks and -6dB for mono. You can experiment with more extreme values for special effects.

Range: -12.00 dB to +12.00 dB, default +6.00 dB

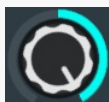


**PANORAMA LEFT**  
100% left

### Panorama left

Panorama left defines the panorama applied to the left channel only.

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



**PANORAMA RIGHT**  
100% right

### Panorama right

Panorama right defines the panorama applied to the right channel only.

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

Volume left

0.00 dB

### Volume left

Volume left defines the input volume adjustment for the left channel.

Range: silence to 12.0 dB, default 0.00 dB

Volume right

0.00 dB

### Volume right

Volume right defines the input volume adjustment for the right channel.

Range: silence to 12.0 dB, default 0.00 dB

Invert left channel

### Invert left channel

Invert left channel inverts the sign of all samples in the left channel. It is useful when processing recordings, that are out-of-phase. It should not be used as a form of stereo-expansion.

Invert right channel

### Invert right channel

Invert right channel inverts the sign of all samples in the right channel. It is useful when processing recordings, that are out-of-phase. It should not be used as a form of stereo-expansion.

Swap left and right

### Swap left and right

Swap left and right swaps samples between the left and right channels.

Silence when playback is ON

### Silence when playback is ON

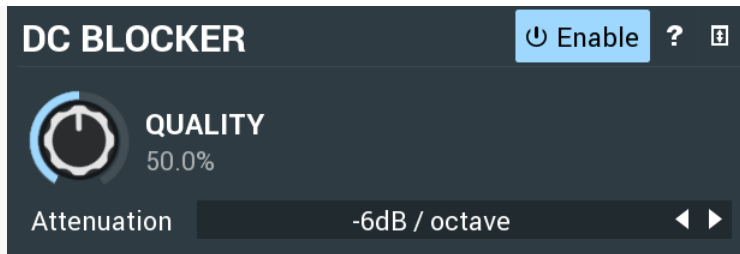
Silence when playback is ON makes the output of the plugin silent when the playback is enabled and can be useful for creating narrated videos for example.

Silence when playback is OFF

### Silence when playback is OFF

Silence when playback is OFF makes the output of the plugin silent when the playback is disabled and can be useful for creating narrated videos for example.

# DC blocker



DC blocker activates a DC blocking filter. DC is a zero-frequency usually caused by analog to digital converters, tube simulations etc. and you can picture it by simply moving the waveform up or down. It can be considered an , because, although it cannot be heard, it lowers the available bit-space, so it may be necessary to remove it. However since it is a filter and you should minimize filtering of the audio, you should use it only when needed, not as a general precaution.

**Enable DC blocker**

DC blocker activates a DC blocking filter. DC is a zero-frequency usually caused by analog to digital converters, tube simulations etc. and you can picture it by simply moving the waveform up or down. It can be considered an , because, although it cannot be heard, it lowers the available bit-space, so it may be necessary to remove it. However since it is a filter and you should minimize filtering of the audio, you should use it only when needed, not as a general precaution.



**Quality**

Quality defines the cut-off frequency of the DC blocker filter. There is no difference in CPU requirements, however higher quality causes a longer filter tail. This means, that after the audio signal finishes, the filter will decay for a longer time, which may be perceived as post-ringing.

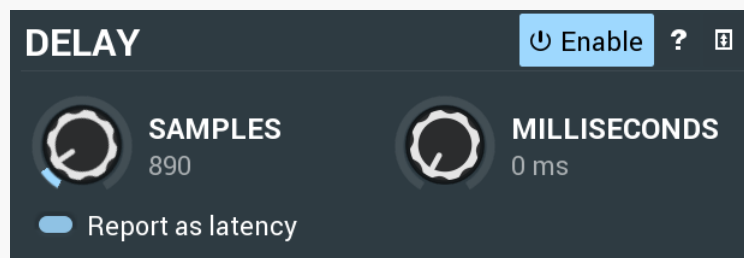
Range: 0.00% to 100.0%, default 50.0%



**Attenuation**

Attenuation defines the attenuation for the DC component. Higher attenuation causes a longer filter tail, a higher effect on the frequencies above DC and a higher CPU cost.

## Delay panel



Delay panel contains multiple ways of delaying the input signal. You may use it to fix phase problems, compensate for latency etc.

**Enable Delay panel**

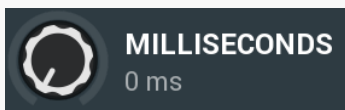
Delay panel contains multiple ways of delaying the input signal. You may use it to fix phase problems, compensate for latency etc.



**Samples**

Samples defines the number of samples by which to delay the signal.

Range: 0 to 10000, default 0



**Milliseconds**

Milliseconds defines the time by which to delay the signal.

Range: 0 ms to 10000 ms, default 0 ms



**Report as latency**

Report as latency switches the delay into "negative delay", which means that the plugin will not actually perform any kind of delay processing, instead it will report the requested delay as latency assuming the host will compensate for it by delaying all other tracks,

hence in a way shifting this track forward in time.

## RMS panel

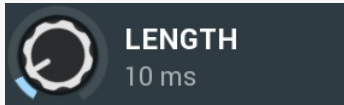


RMS panel contains RMS (root-mean-square) detection parameters. RMS is usually used as the input for envelope detector for dynamic processors.

Enable

### RMS panel

RMS panel contains RMS (root-mean-square) detection parameters. RMS is usually used as the input for envelope detector for dynamic processors.



### Length

Length defines length of the RMS measurement period. A higher value provides smoother results, but a slower response to transients.

Range: 0 ms to 1000 ms, default 10 ms



### Peak hold

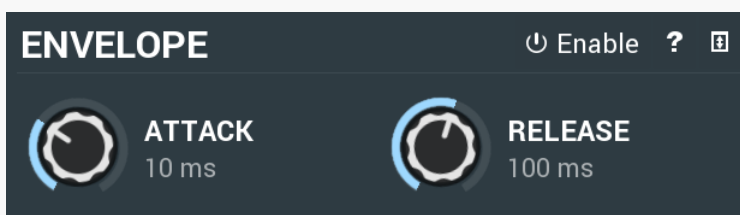
Peak hold defines the time that signal level detector holds its maximum before the release stage is allowed to start. As an example, you can imagine that when an attack stage ends there can be an additional peak hold stage and the level is not yet falling, before the release stage starts. This is true 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

## Envelope panel



Envelope panel provides a simplified envelope detector. It doesn't make sense on its own, but you can use it to prepare a secondary input for a side-chain compressor for example.

Enable

### Envelope panel

Envelope panel provides a simplified envelope detector. It doesn't make sense on its own, but you can use it to prepare a secondary input for a side-chain compressor for example.



**ATTACK**  
10 ms

## 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**  
100 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:

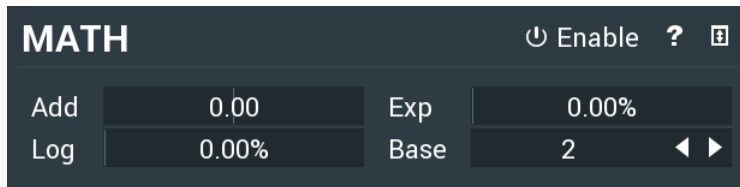
*In a **compressor** the release time controls how quickly the measured level falls below the threshold and the compression stops. As a result a very short release time makes the compressor stop quickly, for example, leaving the sustain of a snare drum intact. On the other hand, a very long release keeps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

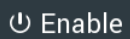
*In a **gate** the situation is similar to a compressor - the release time controls how quickly the measured level can fall below the threshold at which point the gate closes. Having a longer release time in a gate is a perfectly acceptable option. The release time will basically control how much of 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: 0 ms to 1000 ms, default 100 ms



Math panel contains mathematical operations which may come handy for some audio processing.



### Math panel

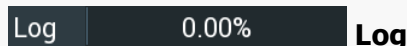
Math panel contains mathematical operations which may come handy for some audio processing.



Add increases every sample by the specified value. Note that this basically adds a DC component.  
Range: -1.00 to 1.00, default 0.00



Exp makes every sample  $y = x ^ \text{base}$  and mixes it with the original value. This operation inverts logarithm, however please note that these operations can be completely inverted only for positive input values.  
Range: 0.00% to 100.0%, default 0.00%

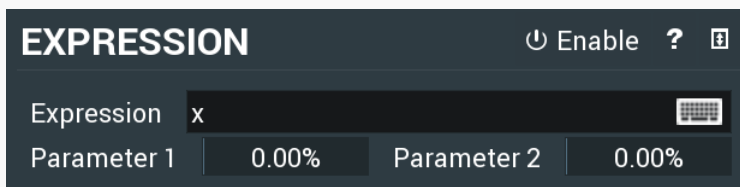


Log makes every sample  $y = \log_{\text{base}}(x)$  and mixes it with the original value. This operation inverts exponential, however note that these operations can be completely inverted only for positive input values. Logarithm returns correct values when used for samples above zero only. For samples lower than or equal to 1E-20, it returns the same value as for 1E-20.  
Range: 0.00% to 100.0%, default 0.00%



Base defines the base for the logarithm and exponential functions.

## Expression panel



Expression panel lets you perform processing based on mathematical expressions. Please note that this processing is much slower than using standard components and also the audio quality may be worse. However it may be useful for specific purposes or testing new ideas.



### Expression panel

Expression panel lets you perform processing based on mathematical expressions. Please note that this processing is much slower than using standard components and also the audio quality may be worse. However it may be useful for specific purposes or testing new ideas.



Expression defines the mathematical expression to use. The expression is calculated on every sample and the output is limited to a range of -1 ... +1 to avoid problems with invalid formulas. Some hosts do not let keyboard events pass in which case you may need to edit the formula in a text editor and then copy & paste it into the plugin. The following variables are provided:

- x** equals the current input sample.
- x1,x2,x3,x4** are the previous input samples.
- y1,y2,y3,y4** are the previous output samples.
- p1,p2** equal parameter values.
- sr** equals the current sampling rate.
- n** equals the index of current sample starting from the initialization.

Expression evaluator uses traditional C/C++ style formatting, which is natural for most people. It provides arithmetics, logical and conditional operators. Following terms are supported:

Unary operators: - (negative sign), ! (logical negation)

Binary arithmetic operators: +, -, \*, /, ^, %, max, min

Binary logical operators: ==, !=, <, <=, >, >=, &&, ||, ^^

Unary functions: log, sqrt, sqr, abs, exp, sin, cos, tan, sinh, cosh, tanh, inv, asin, acos, atan, ln, log10, sgn, floor, ceil,

**round, rand, f01** (frequency from 20...20000 into log scale 0..1)

Ternary logical operator: **a ? b : c** (if a is true, then the result is b, otherwise it is c)

Constants: **pi, e, sqrt2, ln2**

*Example 1: the following expression implements a special case of a biquad filter:*

$0.3*x + 0.2*x1 - 0.3*x2 - 0.1*y1 + 0.4*y2$

*Example 2: the following expression implements a 1000Hz sine wave generator:*

$\sin(1000 * 2 * \pi * n / sr)$

Parameter 1 0.00% **Parameter 1**

Parameter 1 defines the variable **p1** (0..1), which may be used in the expression.

Range: 0.00% to 100.0%, default 0.00%

Parameter 2 0.00% **Parameter 2**

Parameter 2 defines the variable **p2** (0..1), which may be used in the expression.

Range: 0.00% to 100.0%, default 0.00%

## Coder panel

CODER

🔌 Enable ? 📄

Mode

Left+Right -> Mono+Stereo



Coder panel contains operations for encoding and decoding the audio signal between different representations.

🔌 Enable

### Coder panel

Coder panel contains operations for encoding and decoding the audio signal between different representations.

Mode

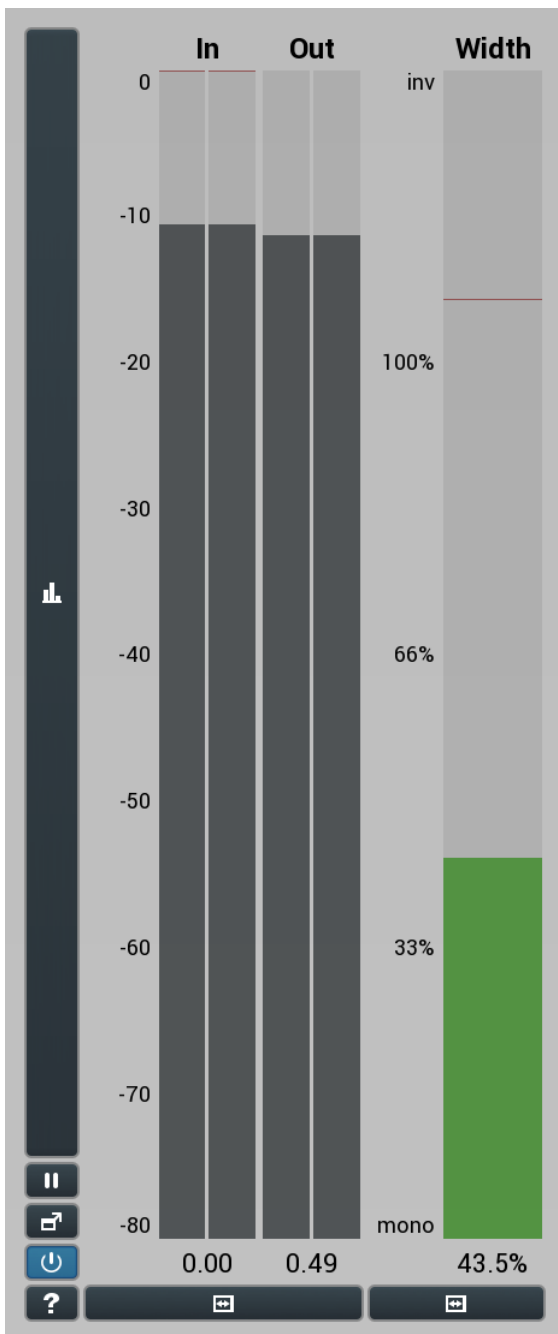
Left+Right -> Mono+Stereo



#### Mode

Mode defines type of the encoding to perform. It lets you convert the input into mid/side representation, so that you can use it in other plugins for example. It also lets you audition the mid, side, left or right channels.





### 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 upsampling 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.

**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 button**

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 button**

Pause button pauses the processing.



### **Popup button**

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 button**

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



### **Collapse button**

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



### **Collapse button**

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



### **Collapse button**

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

