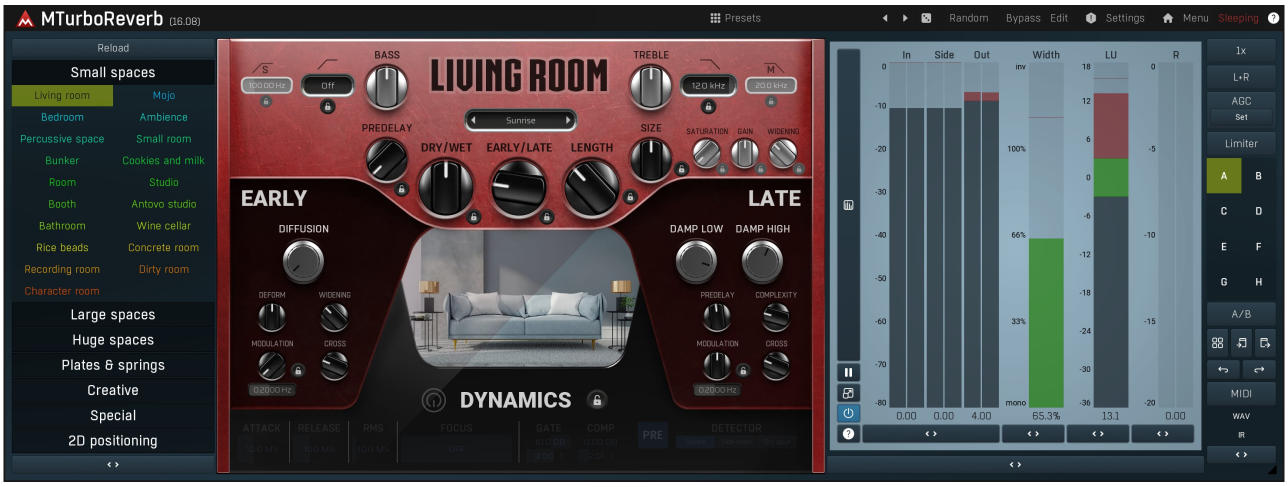
MTurboReverb



Overview

MTurboReverb is probably the most powerful algorithmic reverb ever made. Most reverbs are based around a single algorithm, for which you can change certain properties, such as reverb length. That makes them very limited in terms of use and sound. MTurboReverb comes with currently about 100 reverb designs (based on different, often multiple, algorithms), complex and simple, realistic and creative, large and small, featuring all kinds of advanced factors such as pitch shifting or frequency shifting. So in a way it serves as 100 different reverb plugins (or hardware units) and more are being created.

All these reverbs are available by selecting one of the devices on the main screen (called **Easy screen**). All of them share a very similar GUI, so that you in fact instantly learn to use all of these reverbs just by learning one. You can browse the available reverbs simply by clicking on them on the left side of the Easy screen. Use the **Display locks** button on top of the list to show locks next to most parameters, which let you lock them so they stay the same while browsing the reverbs. For instance, if you are using the reverb as a send, which was quite common mainly to save CPU, you will want to set the **Dry/Wet** to 100%. To do that, just click the lock button next to the Dry/Wet control and it will stay at that value while changing reverbs.

Most of the reverbs feature everything you'd usually need, from **Dry/Wet** and **Length** controls through to parameters of early and late reflections and advanced dynamics processing, all available instantly from the Easy screen and documented below.

Under the hood

MTurboReverb does all that magic by providing a simple programming language that lets you actually define the algorithm that powers the reverb. It has been designed in such a way that creating these algorithms is actually very simple. In fact, each algorithm is defined by a single line! Despite all the versatility the engine is actually extremely CPU efficient. Anyway do not worry; you don't need to do any programming yourself, unless you feel especially creative. The devices will probably keep you busy for a long time.

The **Edit** screen provides access to the underlying functionality and contains several modules. First there are 4 ER (Early Reflection) sections. These are more graphical and let you define the early reflections directly via a tapped delay line. Presets for these sections are available and there are hundreds of predefined ERs. Besides the classic tapped delay line, traditionally used for producing early reflections, each ER sections actually provides additional algorithms such as Diffuser or Room, so that you can actually design the whole reverb just using the ER sections. But that would exploit only a small part of the plugin's potential.

The main power lies in the 6 LR (Late Reflection) sections. So in a way there are up to 6 reverbs running in parallel, and that's not including the ERs, you can only imagine how big the potential is. Each LR section provides an **Algorithm** field, which lets you design the actual processing unit, and define several parameters that apply to the algorithm automatically. These include dampening, size, length and other basic parameters as well as advanced controls defining the underlying system of delays and other structures.

Finally there is a fully-featured dynamics section, that can process the reverb input or output, and 2 dynamic equalizers, one processing standard left/right channels and the other processing mid/side channels. Finally there are modulators at your disposal, if you really aim at being extra creative. And the huge set of multiparameters let you build Easy screen GUIs for your own reverb designs (they have been used to create all the predefined reverbs).

MTurboReverb vs. MTurboReverbLE

MTurboReverb comes in 2 licence editions - MTurboReverb is the full licence, which gives you access to all features of the plugin and also to the multiband version MTurboReverbMB. This full licence is also part of the MTotalFXBundle and MCompleteBundle.

MTurboReverbLE is a limited licence, which provides all the devices on the Easy screen. It doesn't let you use the multiband version or the Edit screen, hence you cannot design your own reverbs. It should however be far more than enough for everyday mixing/mastering/production, it's like 100 reverbs after all.

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).

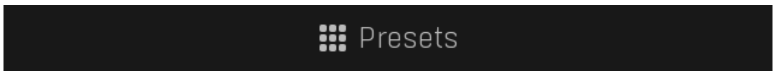
In most cases the devices are highlighted using different text colors. In some cases the colors only mark different types of processing, but in most cases the general rule is that **black/white devices** are the essential ones designed for general use. **Green devices** are designed for a specific task 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.



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.

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

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

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

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

Panic
Panic button resets the plugin state. You can use it to force the plugin to report latency to the host again and to avoid any audio problems. For example, some plugins, having a look-ahead feature, report the size of the look-ahead delay as latency, but it is inconvenient to do that every time the look-ahead changes as it usually causes the playback to stop. After you tweak the latency to the correct value, just click this button to sync the track in time with the others, minimizing phasing artifacts caused by the look-ahead delay mixing with undelayed audio signals in your host. It may also be necessary to restart playback in your host.
Another example is if some malfunctioning plugin generates extremely high values for the input of this plugin. A potential filter may start generating very high values as well and as a result the playback will stop. You can just click this button to reset the plugin and the playback will start again.

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

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

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

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

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

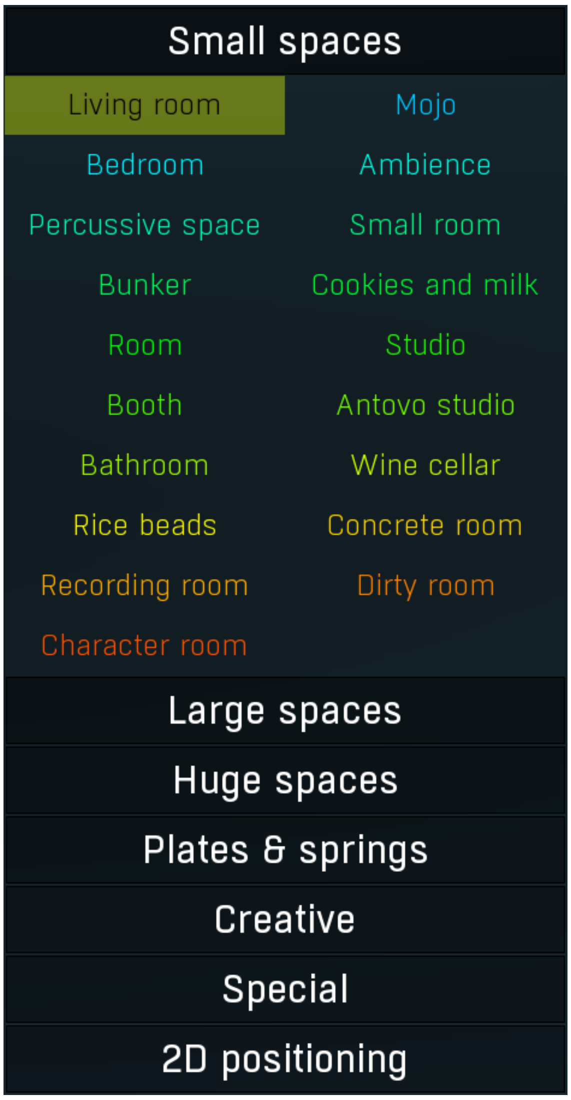
Dry/Wet affects determines, for Multiband plug-ins, which multiband parameters are affected by the Global dry/wet control.

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

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

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

Reload
Reload
Reload button reloads the device and sets all non-locked controls in the current device to their default values. It may be useful, since the plugin stores current settings when switching between the devices, hence this button is a quick and easy way to get the defaults for the devices, before you changed them. If you want to reload all parameters for the device, you must unlock the Easy screen locks or disable them all by turning off the On/Off button in the Global Locking panel in Edit mode.



Device selector

Device selector lets you choose from the predefined devices (previous 'active presets'). These are different from normal presets as they can actually have Easy-mode controls available via knobs or buttons. Click on an device to load it. Check out our video tutorials for information about creating your own devices. Although you cannot put your own devices into this selector, you can still save them as normal presets and on loading they will work in the exactly same way.
When browsing the devices, the plugin stores the control values (multiparameters). It doesn't store the full settings, only the multiparameters, so that if you switch between the devices, your settings will be kept intact, unless you switch to edit screen and perform some advanced editing, in which case it is recommended to use the A-H presets to store your work.

Collapse

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

Comp

Comp controls the threshold for the compressor processing the input or reverberation signals (selected using the **Pre** button in the panel), the minimal signal level, when the effect is applied. Above this level the signal is being compressed. Please note that by 'level' we are talking about the level that the attack/release detector produces, not the peak level for instance. When this parameter is set to its maximum, the compressor is effectively disabled.

Gate

Gate controls the threshold for the gate processing the input or reverberation signals (selected using the **Pre** button in the panel), that is, the minimal signal level below which the effect is applied. Below this level the signal is being gated; with high ratios this essentially means the signal is silenced. Please note that by 'level' we are talking about the level that the attack/release detector produces, not the peak signal level for instance. When this parameter is set to minimum, the gate is effectively disabled.

Pre

Pre switch makes the dynamics module process the input signal as opposed to processing the reverberation signal. Please note that this switch does NOT change the dry input signal being mixed using the global **Dry/Wet** knob, that is the dynamics processing is applied AFTER the signal is tapped for the dry component fed into that knob.

Dynamics panel

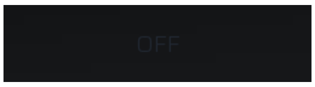
Dynamics panel controls the dynamics processing applied to the input or the reverberation signal (selected using the **Pre** button in the Dynamic Globals panel). This includes gating and compression, both of which are very useful on both signals.

Gating applied to the input signal is a classic technique that lets you reverberate only loud parts of the signal. It is typically used on drums to produce a reverb tail for snare drums for example.

Compressing the input signal lowers the dynamic range creating a "flatter" sound, which creates more stable reverberation without actually flattening the input signal itself.

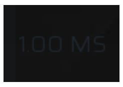
Gating the reverberation signal is a rather specific effect which lets you abruptly terminate the reverberation signal when its level goes below the threshold. When a long reverb is used, it can often muddy the mix during the parts where the particular instrument isn't playing anymore, or create long endings when used on master, which then need to be faded-out by some other method. Using a gate with threshold low enough can provide an automatic solution which stops the reverb tail sooner than it normally would, removing the muddiness / tail.

Compressing the reverberation signal flattens the reverberation signal and when overused can introduce the typical compression character many are seeking on the tracks themselves, but on the reverberation signal instead. It is mainly useful as a creative tool.



Focus

Focus controls the frequency that the dynamics module is listening to the most. This works as a band-pass filter removing unwanted frequencies from the detector. By default the focus is set to the maximum value and the filter is disabled. Lower the value to enable the filter and let the processor focus on the specified frequency. For example, when you want to produce areverberation tail for snare drums in mixed drums, enable the dynamics module, activate the gate (which is activated by default) and set the focus to the fundamental frequency of the snare drum (around 200-500Hz).



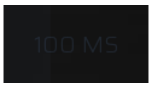
RMS length

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.



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.

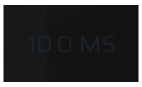
To understand the working of a level detector, it is best to cover the typical cases:

*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.



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.

Comp ratio
Comp ratio controls the ratio for the compressor. The higher the ratio, the stronger the effect is, above the threshold.

Gate ratio
Gate ratio controls the ratio for the gate. The higher the ratio is, the stronger the effect is, below the threshold.

Type

Type lets you choose from several predefined types. These change the sound of the reverb by changing some of the algorithm properties. For instance, a Room device uses Type to choose between several rooms, each of them with more or less different character. Sometimes the differences between types may be huge, sometimes it only changes the stereo field for example; it all depends on the particular device. In some rare cases the control has a different meaning - it is marked Presets and serves as a set of predefined presets, which change several controls of the device. In these cases you need to be careful, since changing the preset changes other parameters, even those you might have already changed yourself.

Treble
Treble controls the high-shelf filter which processes the reverberation signal amplifying or attenuating the high frequency content, giving it more or less "shine". This control is mainly useful for controlling the tone of the reverb. Higher values generally make the reverb brighter, lower values make it darker.

High-cut
High-cut lets you remove highest frequencies using a low-pass filter. This is sometimes useful to clean up the treble, so that the mix doesn't become crowded. In most cases bass and mids are more problematic though.

High-cut mid
High-cut mid lets you filter out highest frequencies using a low-pass filter similarly to **High-cut**, however it only affects the mids, the center signal. Hence it leaves the side signal intact. It is useful during mixing to clean up the treble, without actually removing the treble, since these highest frequencies will stay intact in the sides, where they contribute on the wide reverberation tail. It is usually better to try this control first before using the **High-cut**, which kills the high-end completely.

Length

Length controls the reverb decay time. This parameter affects the late reflections only since the early reflection length is constant and each echo is generated separately. Late reflections are generated in a very different way similar to how it happens in nature - using a system of delays and feedback lines more and more echoes are produced in time with their levels constantly lowering. Mathematically speaking the reverb decay never ends, so in a way an algorithmic reverb is infinite, and the reverb time is defined as the time after which the echoes decay to approximately -60dB, which makes it virtually inaudible in the context of a full scale signal.

Please note that in physical spaces the reverb length naturally relates to the size of the space, which is not true for algorithmic reverbs, which let you specify length and size separately. For that, look at the **Size** parameter in the Late Reflections panel, if provided.

Early/late

Early/late defines the ratio between early and late reflections. Generally early reflections characterize the space and position of the source object and late reflections (reverberation) indicate the room properties. Usually the more audible the early reflections, the closer the dry signal seems and conversely the more late reverberation that is present, the further away it sounds.

Early reflections represent only the initial portion of the reverb tail, when the first few reflections of the sound from the nearest walls reach the ears/head. After some time the sound reflects so many times from various walls that the echo density (the number of echoes per second) becomes very high and the reverb in fact decays into noise (if designed that way). And this is the part known as the late reflections, which usually sounds smooth and can potentially be extremely long, while early reflections only last up to a few hundred milliseconds.

Predelay

Predelay defines the initial delay before the actual response, which simulates the space between the sound source and the listener. The longer the predelay is, the further away the source seems. At some point (around say 100ms) the brain stops understanding that that the reverberation belongs to the dry signal and starts interpreting them separately, and then the dry signal becomes close to the listener again. Predelay can therefore be used to control the distance from the source to the destination, but detaching the 2 signals can also be useful for instance to fill up a mix that isn't full enough without smearing the signal with the reverb itself.

Size

Size controls the size of the virtual space. It is usually implemented by increasing the echo delay size. Please note that unlike real spaces, where higher size usually means longer reverb time, the algorithmic reverberation provides these parameters separately. In effect the size parameter controls mainly how the echoes build up. Higher sizes make the echoes build up mosre slowly (since the distance for the sound to travel from one wall to another is higher). Conversely lower sizes provide faster echo build up, often to the point where the room becomes so small that multiple walls are similar in distance and additional modes start to form and the reverb starts sounding like a small resonant box. In most devices the range for this parameter is intentionally kept large enough for both natural reverberation and the extreme creative effects.

Widening

Widening controls the stereo width of the reverberation signal. The algorithm is fully mono-compatible as it only extends the existing stereo field and no new signal is added. The default 0% means no processing at all. Lowering the value produces narrower results, eventually leading to a monophonic reverb. Higher values produce wider results, however although the width may initially sound pleasing, it may be quite fatiguing, so use it with caution.

It may be of a great help when mixing however, as in these cases it is often advantageous to provide a different width for each instrument in order to provide better separation for the listener.

Bass

Bass controls the low-shelf filter which processes the reverberation signal amplifying or attenuating the low frequency content, giving it more or less "boom". This control is mainly useful for controlling the tone of the reverb.

It is often useful to amplify the bass content in orchestral and contemporary epic music for example, partly to make the reverb sound darker (it might be better to lower the **Treble** though), but always be careful not to muddy the bass content too much. Attenuating the bass content helps during mixing to free the low-end part of the spectrum, however it is usually better to use the **Low-cut** or **Low-cut side** controls instead.

Gain

Gain controls the power amplification applied on the reverberation signal. In most cases the devices are designed so that the reverberation level is similar to the input signal already and hence the **Dry/Wet** control works well.

To check if everything is ideal, set the dry/wet control to 0% and listen / measure the loudness (using the LU meter). Then set the dry/wet to 100%, so that you are auditioning only the reverberation signal and the loudness should be similar. If not, use the gain control to make it similar. When the reverberation signal is similar in loudness to the input signal, the dry/wet control will work better more effectively, since it will avoid the loudness deception, which would normally make you think that the louder signal sound better, even if it that is actually not true.

Saturation

Saturation controls the amount of saturation applied to the reverberation signal. It is an effect that has no equivalent in natural reverberation, but it can highly enhance the output, especially when it comes to the high-end content and brightness. It is recommended to use it with caution however, since analog saturation is technically a smooth distortion and as such may not be very suitable for acoustic music for instance.

Low-cut side

Low-cut side lets you filter out low frequencies using a high-pass filter similarly to **Low-cut**, however it only affects the side signal and keeps the center (mids) intact. This is very useful during mixing to keep the bass content clean. By filtering only the sides you can keep the bass content centred and clean yet still provide some reverberation. It is usually better to try this control first before using the **Low-cut**, which kills the low-end completely.

Low-cut

Low-cut lets you remove lowest frequencies using a high-pass filter. This is often useful for cleaning up the bass content, so that the mix doesn't become crowded / muddy. In fact it is nearly always useful to use either low-cut or **Low-cut side** to clean up the low-end to 100-200Hz, where the reverb only creates mud. In some cases it is even useful to filter out everything below say 500-800Hz, especially for vocals, where the reverb then provides nice smooth high-end, but doesn't interfere with either the bass and mids and keeps the intelligibility of the vocals.

Dry/Wet

Dry/Wet defines the ratio between the dry signal and the produced reverberation signal. If you use the reverb as a send effect, keep this at 100% and put the reverb on a dedicated bus in your host to which you send all the tracks to be processed. This way you control the amount of reverberation by adjusting the levels of the tracks that you send to the bus. If you use the reverb as an insert, use this parameter to control the amount of ambience versus the direct signal. Usually the higher the dry/wet value is, the more distant the audio seems.

Diffusion

Diffusion increases the number of echoes in the early reflection signal providing a smoother early stage of the reverberation signal. It is often useful to increase diffusion for percussive materials, otherwise the initial attack of each hit could sound like "many hits" when the diffusion is too low. However when processing vocals for example, high diffusion is known to reduce the intelligibility; hence it may be useful to turn it down.

Widening

Widening lets you change the stereo width of the early reflections. Early reflections often follow the input signal very closely, so that the brain may not even be able to distinguish between the input signal and these echoes. In most cases they are also rather wide, which may or may not be desired.
If the input signal is mono, you may enjoy the fact that the early reflections produce a nice phattening stereoizing effect. However you may want the initial part of the reverb to have a similar width as the input signal, so that they blend together well.
Also, when mixing it is often desirable for some instruments to be located in the center, so that the mix doesn't become crowded and muddy. This control lets you make the early reflections more centered (or the other way around - more spread out) without changing the late reflections in any way. That way you can make the reverb fit well to the input signal without affecting the nicely wide smooth late reverberation tail.

Modulation

Modulation controls the amount of modulation, which usually varies the input signal pitch and as such changes the reverberation response in time. It often provides a pleasing evolving character at the expense of additional CPU requirements. It also effectively diminishes the possibility of comb filtering resulting in metallic resonances.

Cross

Cross controls how each input channel affects each output channel. When this is set to 0%, then the left output is generated from the left input only and the right output is generated from the right input only. At 50% the left output is generated 67% from the left input and 33% from the right input. And the right output is generated 67% from the right input and 33% from the left input. When this is 100%, both left and right channels contribute equally (50:50) to both left and right outputs. And when you set this to 200%, the channels are exchanged, hence the left output is generated solely from the right input and vice versa.

Modulation rate

Modulation rate controls the speed of the input modulation. The faster it is, the more pronounced is the effect. It can sound very artificial when overused, which can however be exploited for creative effects.

Damp High

Damp High controls the amount of high frequency dampening for the late reflections generator. In nature the air continuously absorbs the higher frequencies and so the signal continuously gets darker and darker. This control defines how quickly that happens.

Dampening is usually implemented as a feedback high-shelf or low-pass filter and the units are presented as decibels or cut-off frequency, depending on what the control changes - the filter's gain, frequency or both. In any case, maximum usually disables the dampening completely, minimum provides maximum dampening.

Damp Low

Damp Low controls the amount of low frequency dampening for the late reflections generator. Low frequencies are often absorbed by various obstacles and so the signal continuously gets weaker and weaker. This control defines how quickly that happens. Although low frequency dampening doesn't happen in nature as much as high frequency dampening, it is very useful for making the mix cleaner.

Dampening is usually implemented as a feedback low-shelf or high-pass filter and the units are presented as decibels or cut-off frequency, depending on what the control changes - the filter's gain, frequency or both. In any case, minimum usually disables the dampening completely, maximum provides maximum dampening.

Complexity

Complexity controls the complexity of the late reflection generator algorithm. It may have very different meaning and effect for different algorithms (devices), in most cases it controls the number of internal delays and feedback paths and as a result it controls the echo density. Therefore usually the higher the complexity, the smoother and denser is the reverberation signal. at the expense of higher CPU consumption. Higher complexity can also avoid metallic resonances (modes).

Cross

Cross controls how each input channel affects each output channel. When this is set to 0%, then the left output is generated from the left input only and the right output is generated from the right input only. At 50% the left output is generated 67% from the left input and 33% from the right input. And the right output is generated 67% from the right input and 33% from the left input. When this is 100%, both left and right channels contribute equally (50:50) to both left and right outputs. And when you set this to 200%, the channels are exchanged, hence the left output is generated solely from the right input and vice versa.

Predelay

Predelay controls additional delay processing only the late reflections. You can use it to control the delay between the early reflections and the late reflections.

One reason to use it is to avoid cancellation between them - since early and late reflection generators are traditionally separate modules, there is a chance they will produce echoes at the same times, which may potentially cancel each other, which significantly lowers the signal punch. This predelay control lets you delay the onset of the late reflections ensuring they won't overlap. In other cases you may however want the late reflections to start as soon as possible (set predelay to 0ms), so that the echoes build up quickly.

The parameter also lets you completely detach the late reflections from the input signal and the early reflections by using very high predelay. These settings however serve mostly for creative purposes.

Modulation rate

Modulation rate controls the speed of the input modulation. The faster it is, the more pronounced is the effect. It can sound very artificial when overused, which can however be exploited for creative effects.

Modulation

Modulation controls the amount of modulation, which usually varies the input signal pitch and as such changes the reverberation response in time. It often provides a pleasing evolving character at the expense of additional CPU requirements. It also effectively diminishes the possibility of comb filtering resulting in metallic resonances.

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.

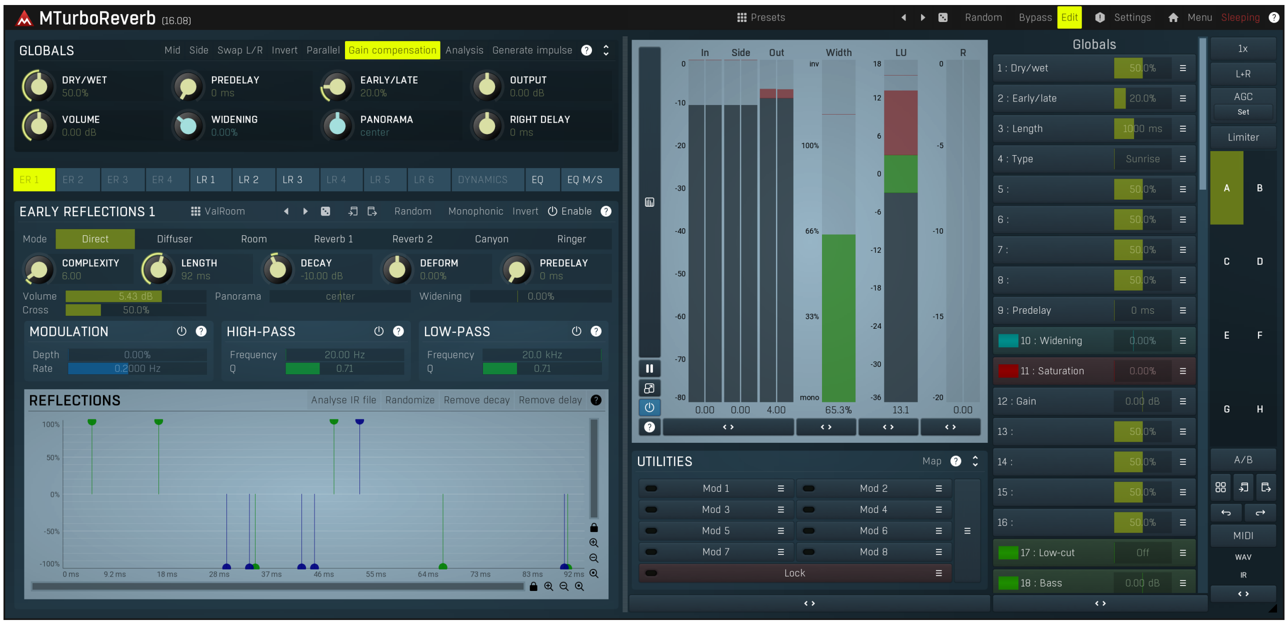
Collapse
Collapse button minimizes or enlarges the panel to release space for other editors.

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

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.

Edit mode



Presets

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.

Random

Randomize

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

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

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

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

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.

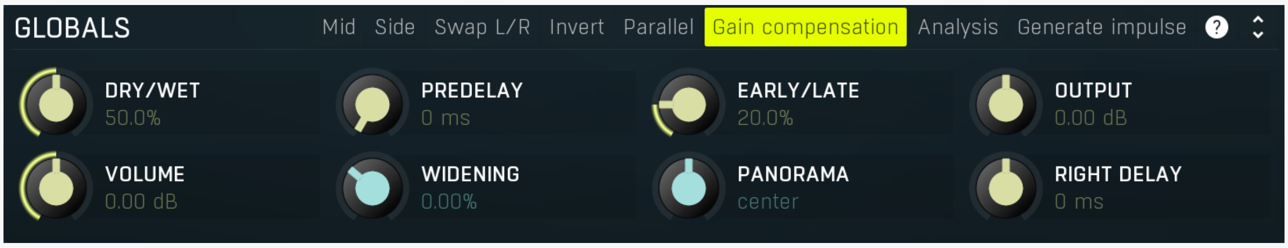
WWW

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.

Globals panel


GLOBALS
Mid Side Swap L/R Invert Parallel Gain compensation Analysis Generate impulse

DRY/WET 50.0%
PREDELAY 0 ms
EARLY/LATE 20.0%
OUTPUT 0.00 dB
VOLUME 0.00 dB
WIDENING 0.00%
PANORAMA center
RIGHT DELAY 0 ms

Globals panel contains basic audio processing features.

Mid

Mid button lets you audition the mid (mono) from the reverberation signal only, that is the mono part of the output. You can use it to check for potential phase cancellations and to get your ears adjusted to the actual center.

Side **Side**
Side button lets you audition the side from the reverberation signal only, that is the stereo part of the output.

Swap L/R **Swap L/R**
Swap L/R button swaps the left and right output channels in the reverberation signal, so it lets you check for stereo field correctness.

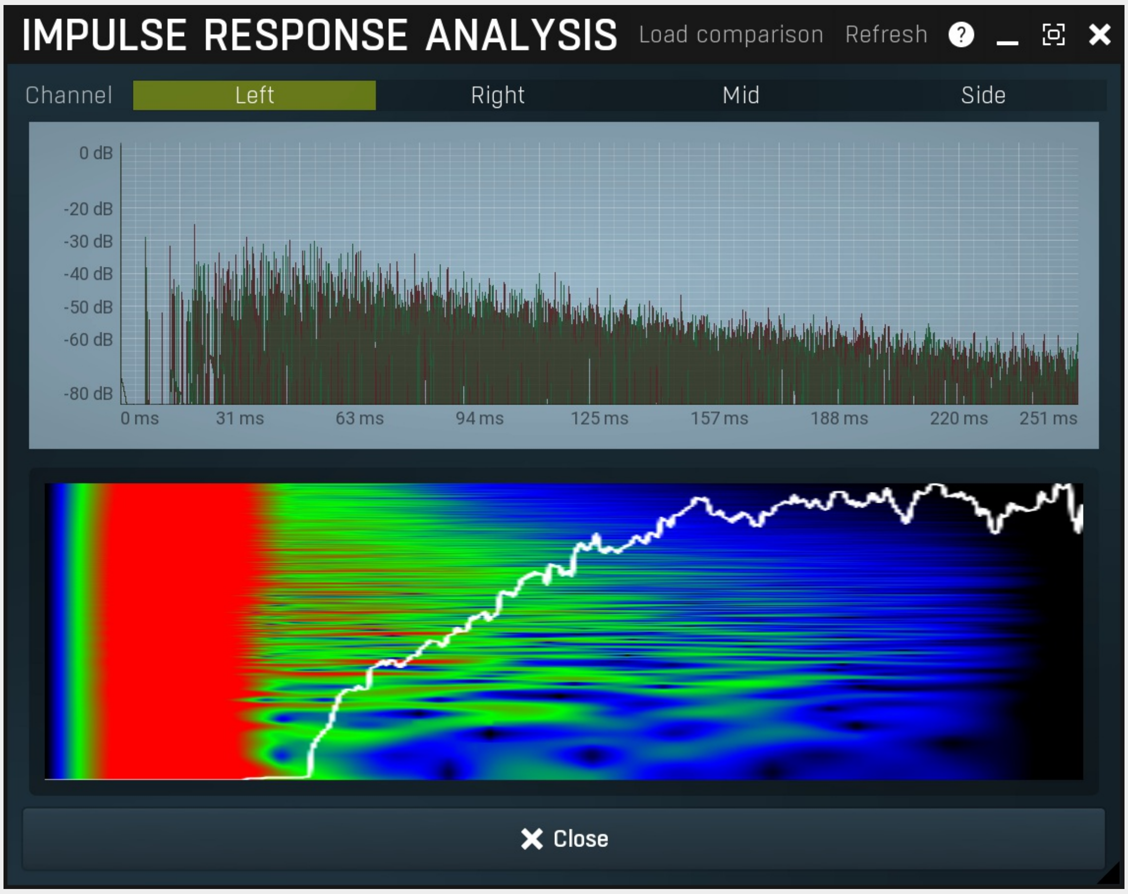
Invert **Invert**
Invert switch inverts the phase of the final reverberation signal. This can be handy when you experience phase cancellation problems when mixing the signal with the input signal using **Dry/Wet** knob for example.

Parallel **Parallel**
Parallel switch defines how the reverb processes individual ER and LR modules and creates the output signal from them. By default this is disabled meaning that the reverb actually works like a hybrid between serial and parallel processing - the signal is taken from the first module to the next one etc. and each of them has **Input** and **Output** parameters, while let you control from where the module takes the signal and how it affects the current output. This eventually lets you create partially serial or parallel processing. However due to the level compensation involved, you may want to disable this, in which case all the modules will be processing the dry input signal and just mixed together to form the final output.

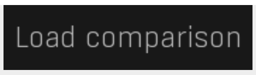
Gain compensation **Gain compensation**
Gain compensation switch activates the gain compensation, which reduces the level when multiple modules are used to maintain approximately constant level. It is enabled by default, since it speeds up the workflow, but in some more complex situations you may want to turn it off, in which case you will have to do all the level handling manually.

Analysis **Analysis**
Analysis button displays a powerful analyzer, which extracts impulse response from the current settings and shows its waveform, spectrum and echo density analysis.

Impulse response analysis

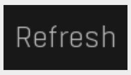


Impulse response analysis provides comprehensive analysis of current reverb settings. It displays the impulse response itself on top. Below you can see the spectrum profile in time and echo density. The analysis automatically updates whenever you change anything related to late reflections, but it doesn't react to all graphs. Note, that in order to sample the impulse response required for the analysis, the main audio processing is temporarily bypassed, so when the analyzer is displayed, actual auditioning may not be very comfortable. However it is very powerful tool for reverb design.



Load comparison

Load comparison button lets you select an external impulse response file and display its analysis next to current analysis view. This can be very helpful when trying to match a different reverb.

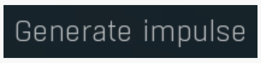


Refresh

Refresh button samples the impulse response of current settings and updates the analysis. This is normally done automatically, but several parameters do not trigger analysis, so you can force the plugin to update using this button.

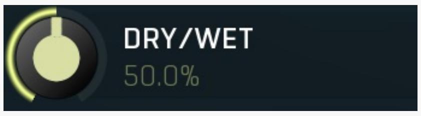
Channel

Channel lets you choose, which channel is analyzed in the bottom analysis view.



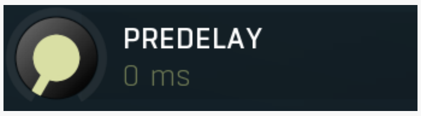
Generate impulse

Generate impulse button makes the reverb produce and process one impulse. It is useful when designing new reverbs.



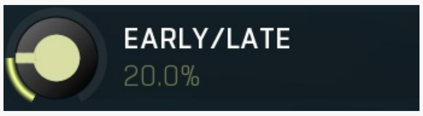
Dry/Wet

Dry/Wet defines the ratio between the dry signal and the produced reverberation signal. If you use the reverb as a send effect, keep this at 100% and put the reverb on a dedicated bus in your host to which you send all the tracks to be processed. This way you control the amount of reverberation by adjusting the levels of the tracks that you send to the bus. If you use the reverb as an insert, use this parameter to control the amount of ambience versus the direct signal. Usually the higher the dry/wet value is, the more distant the audio seems.
Range: 0.00% to 100.0%, default 50.0%



Predelay

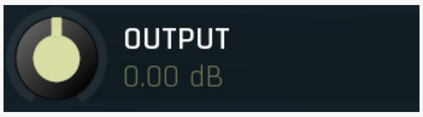
Predelay defines the initial delay before the actual response, which simulates the space between the sound source and the listener. The longer the predelay is, the further away the source seems. At some point (around say 100ms) the brain stops understanding that that the reverberation belongs to the dry signal and starts interpreting them separately, and then the dry signal becomes close to the listener again. Predelay can therefore be used to control the distance from the source to the destination, but detaching the 2 signals can also be useful for instance to fill up a mix that isn't full enough without smearing the signal with the reverb itself.
Range: 0 ms to 1000 ms, default 0 ms



Early/late

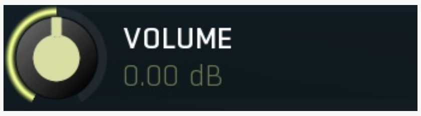
Early/late defines the ratio between early and late reflections. Generally early reflections characterize the space and position of the source object and late reflections (reverberation) indicate the room properties. Usually the more audible the early reflections, the closer the dry signal seems and conversely the more late reverberation that is present, the further away it sounds.

Early reflections represent only the initial portion of the reverb tail, when the first few reflections of the sound from the nearest walls reach the ears/head. After some time the sound reflects so many times from various walls that the echo density (the number of echoes per second) becomes very high and the reverb in fact decays into noise (if designed that way). And this is the part known as the late reflections, which usually sounds smooth and can potentially be extremely long, while early reflections only last up to a few hundred milliseconds.
Range: 0.00% to 100.0%, default 50.0%



Output gain

Output gain defines the gain performed on the reverberation signal. You can use it to match the input level, so that **Dry/Wet** becomes easy to use and won't fool your ears by changing the output loudness.
Range: -24.00 dB to +24.00 dB, default 0.00 dB



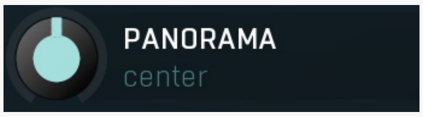
Volume

Volume defines the gain performed on the reverberation signal. It's similar to **Output gain**, but has a different range, which is compatible with the volume parameters of all ER and LR modules. This compatibility can be exploited for some very advanced techniques involving automatic loudness compensation in multiparameter banks.
Range: silence to 12.0 dB, default 0.00 dB

Widening

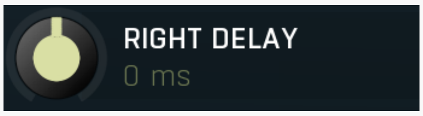
Widening defines the broad-band stereo field widening depth. The algorithm is fully mono-compatible as it only extends the existing stereo field and no new signal is added. This parameter should only be used to control the existing stereo field.

Widening converts the audio into its mid (mono) and side channels, leaving the mid intact and applying a gain to the side channel, then converts the signal back to left and right channels. As a result the stereo image becomes wider (for widening above 0%) or narrower (for widening below 0%). This method of widening the stereo image may initially sound pleasing, however it can quickly become fatiguing on the ear and often sounds unnatural, especially for larger amounts of widening. Use this parameter to control the existing stereo field and as a special effect. Use it to increase width only with caution.
Please note that this algorithm is applied to the reverberation signal only. This is one of the ways to push the audio further away (lower values) or closer (higher values). The advantage is that the original signal's width is kept intact, yet the brain is often able to understand the distance clues.
Range: Mono to 200.0%, default 0.00%



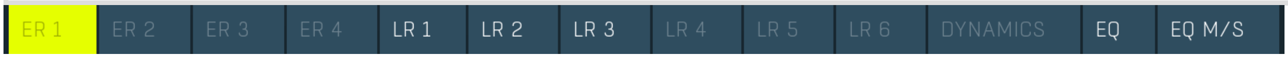
Panorama

Panorama defines the panorama applied to the reverberation signal.
Range: 100% left to 100% right, default center



Right delay

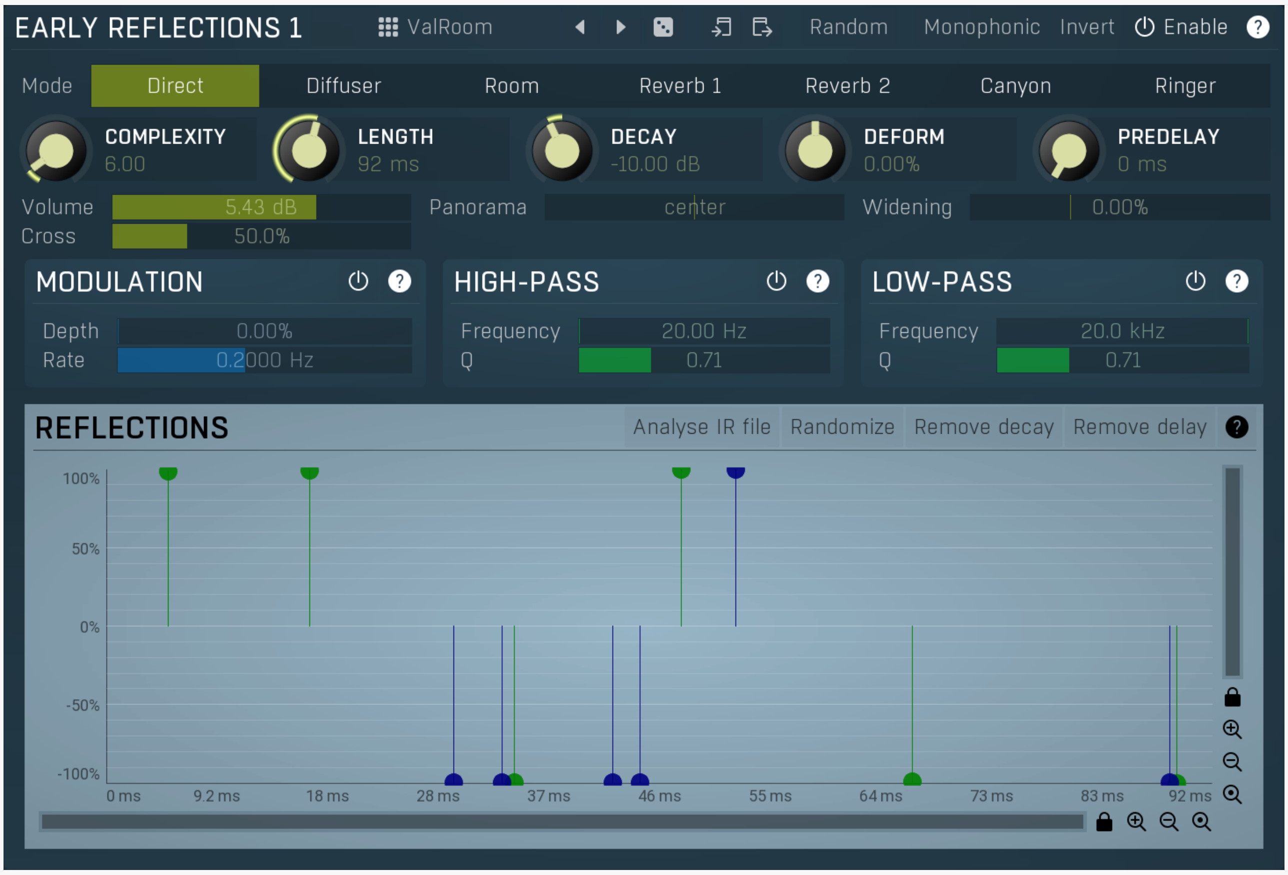
Right delay lets you delay the right channel reverberation signal against the left channel or vice versa. It can be handy for improving the natural stereo field. If the value is positive, right signal is delayed. If it is negative, left channel is delayed instead.
Range: -100 ms to 100 ms, default 0 ms



Tab selector

Tab selector switches between subsections.

Early panel



COMPLEXITY 6.00
LENGTH 92 ms
DECAY -10.00 dB
DEFORM 0.00%
PREDELAY 0 ms
Volume 5.43 dB
Cross 50.0%
Panorama center
Widening 0.00%
Depth 0.00%
Rate 0.2000 Hz
Frequency 20.00 Hz
Q 0.71
Frequency 20.0 kHz
Q 0.71

Early panel contains parameters of the Early reflections generator. It is based on a complex delay system, which you can access directly, or you can let the plugin generate the features automatically, analyzer an impulse response file etc. Early reflections describe the room properties and their purpose is usually to give some initial punch and width to the sound. It is often complicated to set them up manually, as the taps need to follow specific rules, otherwise strong artifacts, typically comb-filtering, may occur, therefore it is often advantageous to let the plugin generate them automatically.

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

Left arrow
Left arrow button loads the previous preset.

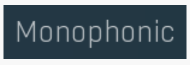
Right arrow
Right arrow button loads the next preset.

Randomize
Randomize button loads a random preset.

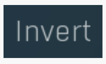
Copy
Copy button copies the settings onto the system clipboard.

Paste
Paste button loads the settings from the system clipboard.

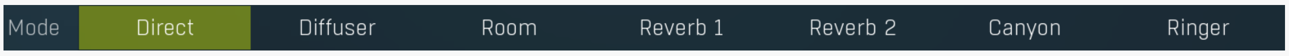
Random
Random button generates random settings using the existing presets.



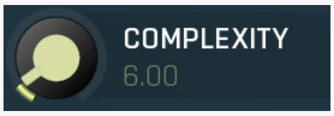
Monophonic
Monophonic button switches to/from single channel mode, where the reflections graph contains only one set of taps and these are used for all channels. In such mode no stereo expansion will occur and usually the stereo width is actually lowered.



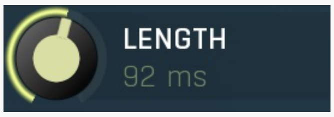
Invert
Invert switch inverts the phase of the signal from this module. This can be handy when you experience phase cancellation problems when mixing the signal with the input signal using **Dry/Wet** knob for example.



Mode
Mode defines the algorithm of the Early Reflections (ER) generator. The whole ER section is generally designed for use in the default Direct mode, but the other modes may be useful creatively.
Direct is the typical algorithm which most reverbs use and that is a tapped delay. The effect of this mode is the least prominent as there is no feedback whatsoever and hence it is also the safest (less likely to produce unwanted muddiness). It is also the least CPU demanding. However, a common problem is that if you want to provide some sort of diffusion even for the early reflections, you may need a lot of reflections to do that, in which case lots of CPU power may be necessary. In these cases it is useful to combine multiple ER generators; for example, one in the Direct mode and another one in Diffuser mode.
Diffuser mode usually requires just a few reflections and the internal feedback can provide the typical diffusive results without any coloration of the sound spectrum. It is characterized by the long initial attack and ringing if either the length is too low or too many reflections are used. Hence it is usually recommended to use short **Length** (e.g. 20ms) and small **Complexity** (say 3-5) values. The diffuser mode is usually used in conjunction with other modes to get a higher density.
Room, **Reverb 1** and **Reverb 2** modes implement some basic reverberation algorithms, which actually sound more like late reflections, so in a way this may be all that you need to design a full reverb. These algorithms usually suffer from low echo density and there's no direct way to control the reverberation time. They may be well used creatively though.
Canyon is rather creative and its decay times are often extremely long.
Ringer implements a serial comb filter system, which has a very characteristic metallic sound character; it is available only for the sake of completeness as it can be useful creatively.



Complexity
Complexity controls the number of early reflections. The more reflections, the more CPU it requires, but the more dense the early reflections can be. With modes other than **Direct** it is often enough to have just a few reflections.
Range: 1.00 to 64.00, default 30.00



Length
Length controls the length of the early reflection interval, basically the time until the last early reflection arives to the listener. It generally controls the room size, which is especially advantageous since it is independent on the late reflection time, hence you can create a big room with short ambience for example.

Range: 1.0 ms to 1000 ms, default 100 ms

Decay

Decay controls the level of the reflections with increasing time. By lowering the decay you can make the reflections, which arrive later, sound less prominent, which is natural, however it also shortens them early reverberation section. By increasing the decay towards 0dB you can make all reflections similar in loudness, providing less natural sound, which can however provide some sort of tightness often useful when mixing.
Range: -60.00 dB to +60.00 dB, default -10.00 dB

Deform

Deform skews the delay times, which in effect changes the character/shape of the virtual room.
Range: -100.0% to 100.0%, default 0.00%

Predelay

Predelay defines the initial delay before the generator response, which simulates the space between the sound source and the listener. There is also a global predelay, which delays the whole output of the plugin generated signal. This one delays only the output of this early reflections generator however and can be used to control the timing of each generator.
Range: 0 ms to 1000 ms, default 0 ms

Volume

Volume defines the level of the output for this Early reflections generator.
Range: silence to 12.0 dB, default 0.00 dB

Panorama

Panorama defines the panorama applied to the Early reflections generator output.
Range: 100% left to 100% right, default center

Widening

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

Widening converts the audio into its mid (mono) and side channels, leaving the mid intact and applying a gain to the side channel, then converts the signal back to left and right channels. As a result the stereo image becomes wider (for widening above 0%) or narrower (for widening below 0%). This method of widening the stereo image may initially sound pleasing, however it can quickly become fatiguing on the ear and often sounds unnatural, especially for larger amounts of widening. Use this parameter to control the existing stereo field and as a special effect. Use it to increase width only with caution.
Range: Mono to 200.0%, default 0.00%

Cross

Cross controls how each input channel affects each output channel. When this is set to 0%, then the left output is generated from the left input only and the right output is generated from the right input only. At 50% the left output is generated 67% from the left input and 33% from the right input. And the right output is generated 67% from the right input and 33% from the left input. When this is 100%, both left and right channels contribute equally (50:50) to both left and right outputs. And when you set this to 200%, the channels are exchanged, hence the left output is generated solely from the right input and vice versa.
Range: 0.00% to 200.0%, default 0.00%

Modulation panel

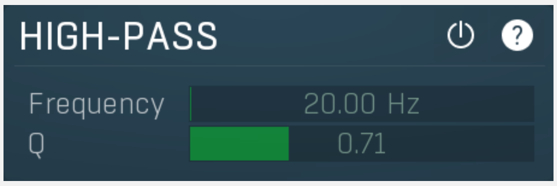
Modulation panel contains the moduation parameters for this generator. When this is disabled, the processor is taking the input signal directly. If you enable it, the input signal is modulated first, which often provides a pleasing evolving character at the expense of additional CPU requirements. It also effectively diminishes the possibility of comb filtering.

Depth

Depth controls the depth of the input modulation useful to further improve the realisticity of the produced reverberation. It can sound very artificial when overused.
Range: 0.00% to 100.0%, default 50.0%

Rate
Rate controls the speed of the input modulation. The faster it is, the more pronounced the effect is. It can sound very artificial when overused.
Range: 0.0100 Hz to 10.00 Hz, default 0.2000 Hz

High-pass panel

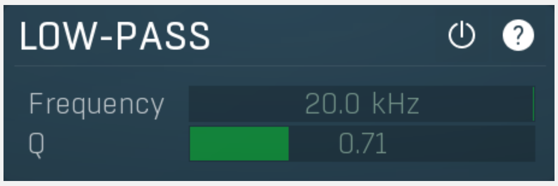


High-pass panel contains parameters of the optional high-pass filter processing the output of this generator.

Frequency
Frequency defines the filter cut-off frequency.
Range: 20.00 Hz to 20.0 kHz, default 20.00 Hz

Q
Q defines resonance of the filter.
Range: 0.05 to 100.00, default 0.71

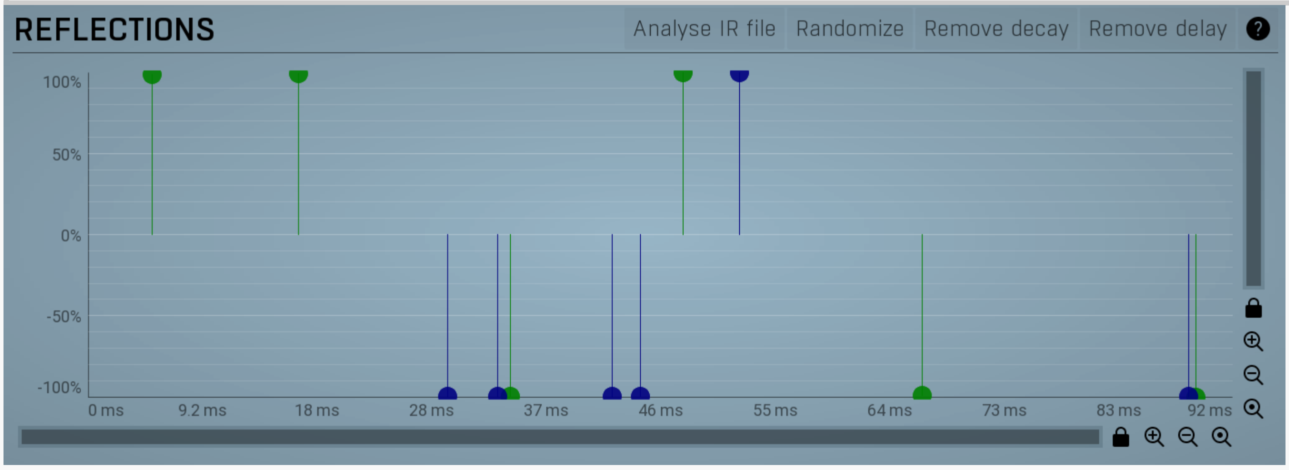
Low-pass panel



Low-pass panel contains parameters of the optional low-pass filter processing the output of this generator.

Frequency
Frequency defines the filter cut-off frequency.
Range: 20.00 Hz to 20.0 kHz, default 20.0 kHz

Q
Q defines resonance of the filter.
Range: 0.05 to 100.00, default 0.71



Reflections editor
Reflections editor contains all the reflections the generator is based on. Each point defined one reflection. Only used reflections are displayed, hence the graph depends on the **Complexity** parameter. Horizontal axis defines time, left being zero delay, right being maximum delay controller by **Length** parameter. Vertical axis contains the reflection level, its meaning depends on the current **Mode**. In the default **Direct** mode it controls the level or loudness of the reflection.

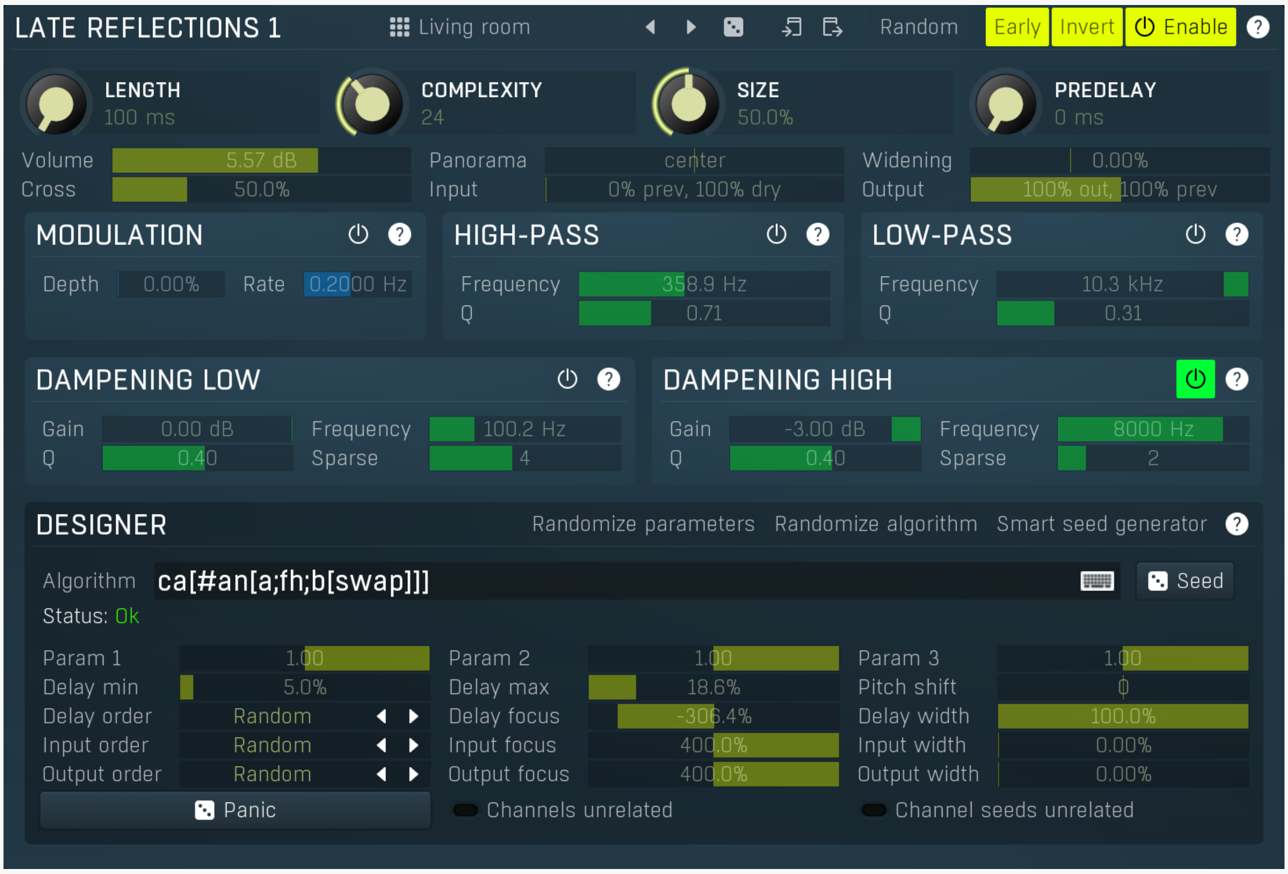
Analyse IR file
Analyse IR file lets you select an impulse respnse file to load and analyze. The plugin will set the reflections according to the prominent reflections in the beginning of that file.

Randomize
Randomize generates a new set of reflections using a specially designed randomization algorithm. Hold **ctrl** to randomly change the current values slightly instead of a full randomization.

Remove decay
Remove decay readjusts the taps to minimize the decay. It is often useful after analyzing an IR file, since despite the results may be convincingly reproducing the original, one cannot really use **Decay** parameter well anymore, since there is already some decay in the taps. Also, early reflections without too much decay are often quite advantageous to disperse and tighten the audio.

Remove delay
Remove delay removes the empty space in front of the first tap and sets the **Length** and **Predelay** appropriately.

Late panel


LATE REFLECTIONS 1 — Living room — Random — Early — Invert — Enable

LENGTH 100 ms
COMPLEXITY 24
SIZE 50.0%
PREDELAY 0 ms

Volume 5.57 dB
Cross 50.0%
Panorama center
Input 0% prev, 100% dry
Widening 0.00%
Output 100% out, 100% prev

MODULATION
Depth 0.00%
Rate 0.2000 Hz

HIGH-PASS
Frequency 358.9 Hz
Q 0.71

LOW-PASS
Frequency 10.3 kHz
Q 0.31

DAMPENING LOW
Gain 0.00 dB
Q 0.40
Frequency 100.2 Hz
Sparse 4

DAMPENING HIGH
Gain -3.00 dB
Q 0.40
Frequency 8000 Hz
Sparse 2

DESIGNER — Randomize parameters — Randomize algorithm — Smart seed generator
Algorithm ca[#an[a;fh;b[swap]]] — Seed
Status: Ok

Param 1 1.00
Delay min 5.0%
Delay order Random
Input order Random
Output order Random
Param 2 1.00
Delay max 18.6%
Delay focus -306.4%
Input focus 400.0%
Output focus 400.0%
Param 3 1.00
Pitch shift 0
Delay width 100.0%
Input width 0.00%
Output width 0.00%
Panic — Channels unrelated — Channel seeds unrelated

Late panel contains parameters of the late reflections generator. It implements a state of the art algorithmic generator system, which you can actually program yourself, but in most cases it is best to start from presets. It lets you define your own reverberation algorithm from the several predefined modules. As all algorithmic reverberators it is based on a very complex feedback delay system and instead of providing direct access to the many parameters it has, it is controlled using a set of pseudorandom number of generators, which configure these parameters automatically. This way you can simply focus on the sound while clicking the randomizer buttons.

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

Left arrow
Left arrow button loads the previous preset.

Right arrow
Right arrow button loads the next preset.

Randomize
Randomize button loads a random preset.

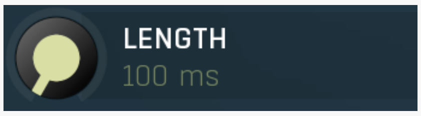
Copy
Copy button copies the settings onto the system clipboard.

Paste
Paste button loads the settings from the system clipboard.

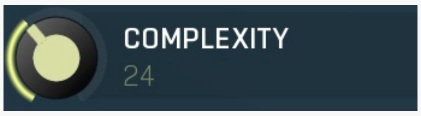
Random
Random button generates random settings using the existing presets.

Early
Early switch makes the plugin think this LR is actually an early reflections generator. It makes it work well with the **Early/late** parameter. Without it you could still use a multiparameter to simulate proper early/late mix, but this option makes it much easier.

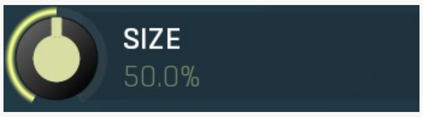
Invert
Invert switch inverts the phase of the signal from this module. This can be handy when you experience phase cancellation problems when mixing the signal with the input signal using **Dry/Wet** knob for example.



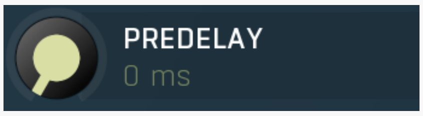
Length
Length controls the decay length, an approximate time until the reverb decays to -60dB. Note that the actual decay time may vary with several other settings. Also the **Algorithm** may highly affect the decay length.
Range: 100 ms to 60000 ms, default 2000 ms



Complexity
Complexity controls the structure of this late reflections generator. Its meaning depends on the **Algorithm**, but in most cases the higher the value is, the more complex and likely more natural the output will be, however the CPU requirements will increase as well.
Range: 1 to 64, default 8



Size
Size controls the virtual room size. It controls the delay lengths used in the generator. The actual meaning depends on the **Algorithm**.
Range: 0.00% to 100.0%, default 50.0%



Predelay
Predelay defines the initial delay before the generator response, which simulates the space between the sound source and the listener. The longer the predelay is, the further away the source seems. There is also a global predelay, which delays the whole output of the plugin generated signal. This one delays only the output of this generator however.
Range: 0 ms to 1000 ms, default 0 ms

Volume
Volume defines the level of the output for this late reflections generator.
Range: silence to 12.0 dB, default 0.00 dB

Panorama
Panorama defines the panorama applied to the late reflections generator output.
Range: 100% left to 100% right, default center

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

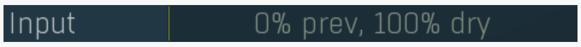
Widening converts the audio into its mid (mono) and side channels, leaving the mid intact and applying a gain to the side channel, then converts the signal back to left and right channels. As a result the stereo image becomes wider (for widening above 0%) or narrower

(for widening below 0%). This method of widening the stereo image may initially sound pleasing, however it can quickly become fatiguing on the ear and often sounds unnatural, especially for larger amounts of widening. Use this parameter to control the existing stereo field and as a special effect. Use it to increase width only with caution.
Range: Mono to 200.0%, default 0.00%

Cross 50.0%

Cross

Cross controls how each input channel affects each output channel. When this is set to 0%, then the left output is generated from the left input only and the right output is generated from the right input only. At 50% the left output is generated 67% from the left input and 33% from the right input. And the right output is generated 67% from the right input and 33% from the left input. When this is 100%, both left and right channels contribute equally (50:50) to both left and right outputs. And when you set this to 200%, the channels are exchanged, hence the left output is generated solely from the right input and vice versa.
Range: 0.00% to 200.0%, default 0.00%


Input 0% prev, 100% dry

Input

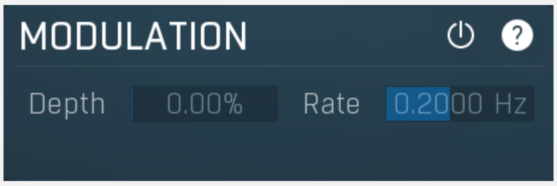
Input controls how much of the original (dry) input and how much of the signal from the previous generators is mixed to form the input for this generator.
If you set this to **0% prev, 100% dry**, the generator will be processing the dry input, hence it will be working in parallel with the previous generators.
If you set this to **100% prev, 100% dry**, then this generator will process an equal mix of both the dry input and the output from the previous section.
If you set this to **100% prev, 0% dry**, then this generator will work in series with the previous generator. Serial processing can produce high density, however it will also increase the risk of comb filtering, which is usually not desired.
Range: 0% prev, 100% dry to 100% prev, 0% dry, default 0% prev, 100% dry

Output 100% out, 100% prev

Output

Output controls the mix ratio between current signal produced by previous generators and output from this one. It is often useful in conjunction with **Input** as in this way you can place all generators serially after each other, as opposed to parallel processing, which is the default.
If you set this to **0% out, 100% prev**, the generator will be effectively doing nothing.
If you set this to **100% out, 100% prev**, then this generator will be fully mixed with the previous output, hence it will be working in parallel with other generators.
If you set this to **100% out, 0% prev**, then this generator's output will replace the current reverberation signal, hence it will work in series with the previous generators. Serial processing can provide high density, however it will also increase the risk of comb filtering, which is usually not desired.
Range: 0% out, 100% prev to 100% out, 0% prev, default 100% out, 100% prev

Modulation panel


MODULATION
Depth 0.00%
Rate 0.2000 Hz

Modulation panel contains the moduation parameters for this generator. When this is disabled, the processor is taking the input signal directly. If you enable it, the input signal is modulated first, which often provides a pleasing evolving character at the expense of additional CPU requirements. It also effectively diminishes the possibility of comb filtering. Note that you can perform additional modulations within the actual **Algorithm**.

Depth 0.00%

Depth

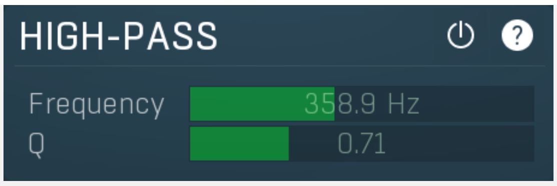
Depth controls the depth of the input modulation useful to further improve the realisticity of the produced reverberation. It can sound very artificial when overused.
Range: 0.00% to 100.0%, default 50.0%

Rate 0.2000 Hz

Rate

Rate controls the speed of the input modulation. The faster it is, the more pronounced the effect is. It can sound very artificial when overused.
Range: 0.0100 Hz to 10.00 Hz, default 0.2000 Hz

High-pass panel



High-pass panel contains parameters of the optional high-pass filter processing the output of this generator.

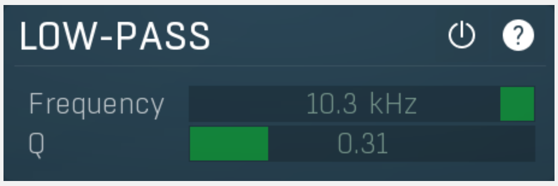
Frequency

Frequency defines the filter cut-off frequency.
Range: 20.00 Hz to 20.0 kHz, default 20.00 Hz

Q

Q defines resonance of the filter.
Range: 0.05 to 100.00, default 0.71

Low-pass panel



Low-pass panel contains parameters of the optional low-pass filter processing the output of this generator.

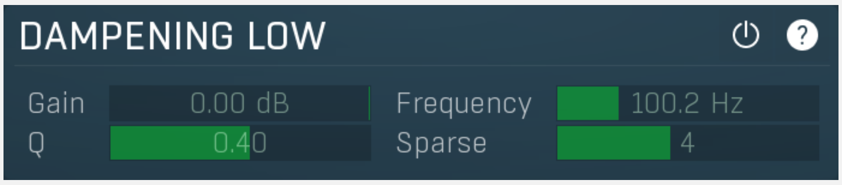
Frequency

Frequency defines the filter cut-off frequency.
Range: 20.00 Hz to 20.0 kHz, default 20.0 kHz

Q

Q defines resonance of the filter.
Range: 0.05 to 100.00, default 0.71

Dampening Low panel



Dampening Low panel contains parameters of the low frequency dampening filter. Its purpose is to simulate absorption by air and walls. If and how is this feature used depends on the **Algorithm**.

Gain

Gain controls the filter gain, hence amount of the dampening. The lower it is, the more of the bass content is removed and less muffled the sound will be.
Range: -20.00 dB to 0.00 dB, default -3.00 dB

Frequency

Frequency controls the filter frequency. The higher it is, the more of the bass content is removed and less muffled the sound will be.
Range: 20.00 Hz to 20.0 kHz, default 200.0 Hz

Q

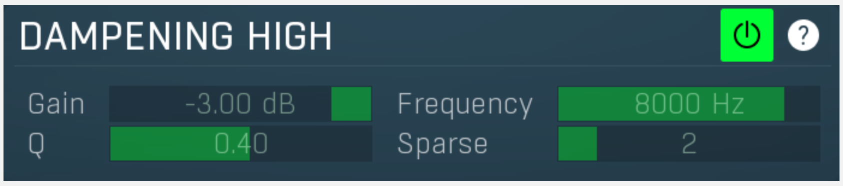
Q controls the filter Q. The lower it is the smoother the effect will be, but it will also be less focused the low frequencies desired to attenuate.
Range: 0.05 to 0.71, default 0.40

Sparse

Sparse lets you avoid filtering all internal delaylines and save CPU. The effect of dampening is lowered this way, but it is in fact more natural as it sort-of simulates different reflection surfaces. Value 1 means that every delay is filtered and provides highest

dampening and highest CPU usage. Value 2 that every second delay is filtered etc. Note that this parameter is supported only by **R** and **RD** algorithm modules.
Range: 1 to 8, default 4

Dampening High panel


DAMPENING HIGH

Gain -3.00 dB
Q 0.40
Frequency 8000 Hz
Sparse 2

Dampening High panel contains parameters of the high frequency dampening filter. Its purpose is to simulate absorption by air and walls. If and how is this feature used depends on the **Algorithm**.

Gain

Gain controls the filter gain, hence amount of the dampening. The lower it is, the darker the sound will be.
Range: -20.00 dB to 0.00 dB, default -3.00 dB

Frequency

Frequency controls the filter frequency. The lower it is, the darker the sound will be.
Range: 20.00 Hz to 20.0 kHz, default 8000 Hz

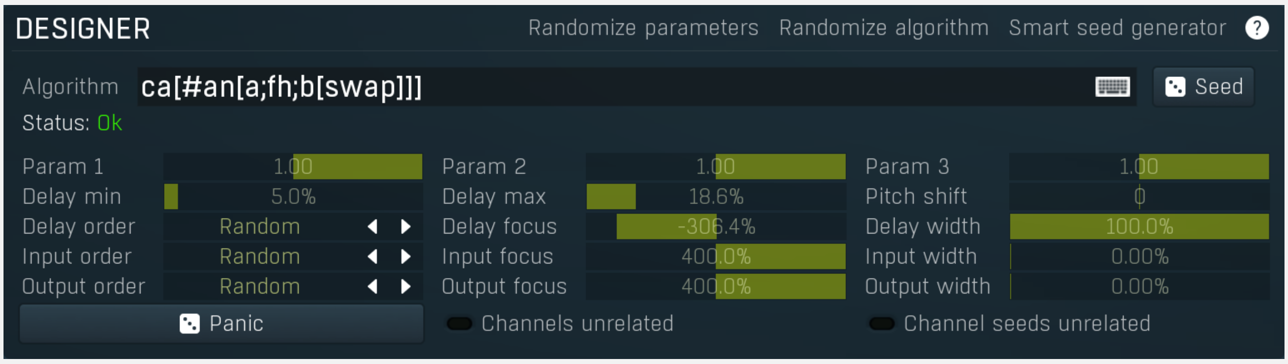
Q

Q controls the filter Q. The lower it is the smoother the effect will be, but it will also be less focused the high frequencies desired to attenuate.
Range: 0.05 to 0.71, default 0.40

Sparse

Sparse lets you avoid filtering all internal delaylines and save CPU. The effect of dampening is lowered this way, but it is in fact more natural as it sort-of simulates different reflection surfaces. Value 1 means that every delay is filtered and provides highest dampening and highest CPU usage. Value 2 that every second delay is filtered etc.Note that this parameter is supported only by **R** and **RD** algorithm modules.
Range: 1 to 8, default 2

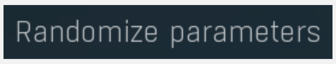
Designer panel


DESIGNER Randomize parameters Randomize algorithm Smart seed generator

Algorithm ca[#an[a;fh;b[swap]]] Seed
Status: Ok

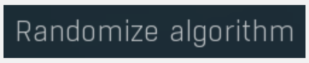
Param 1 1.00
Delay min 5.0%
Delay order Random
Input order Random
Output order Random
Panic
Param 2 1.00
Delay max 18.6%
Delay focus -306.4%
Input focus 400.0%
Output focus 400.0%
Channels unrelated
Param 3 1.00
Pitch shift 0
Delay width 100.0%
Input width 0.00%
Output width 0.00%
Channel seeds unrelated

Designer panel controls the algorithm used in the late reflections generator and its parameters.



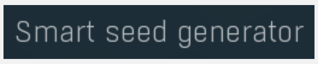
Randomize parameters

Randomize parameters button randomizes all parameters of the algorithm except for the algorithm itself and **Complexity**. It is useful for testing the algorithm. It doesn't always change all parameters, it just changes them somehow. If you hold **Ctrl**, the parameters will only be slightly modified rather than completely randomized.



Randomize algorithm

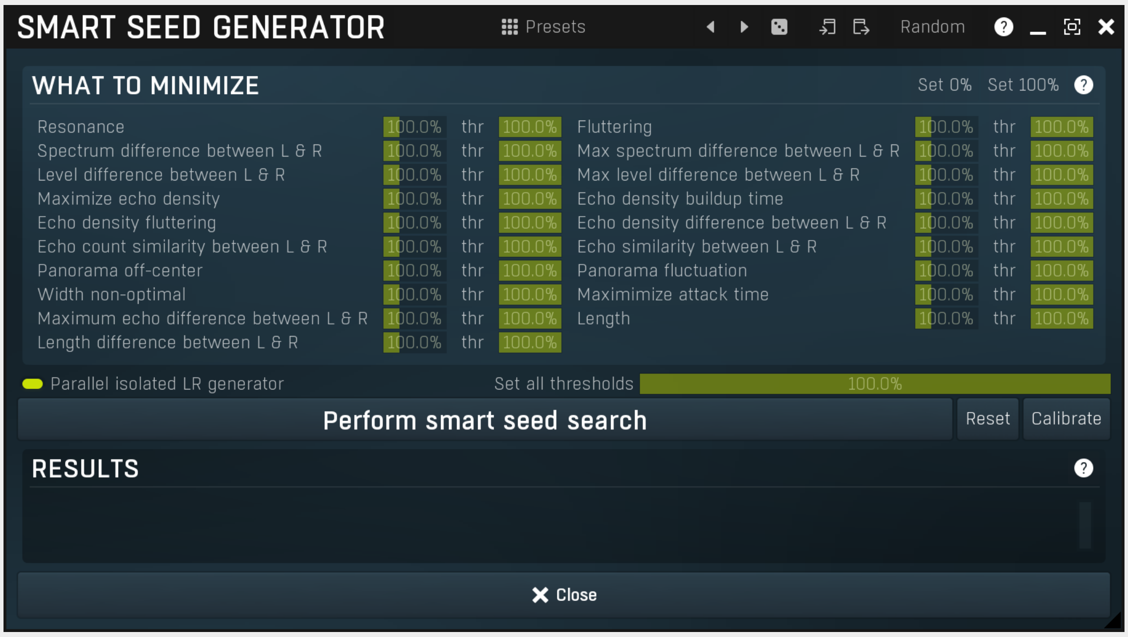
Randomize algorithm button is an experimental algorithm generator. When using this feature, it is highly recommended to keep safety limiter enabled.



Smart seed generator

Smart seed generator button lets you explore thousands of seeds and qualify them automatically, so that you can browse only the best ones. Instead of listening to the sound each seed produces, the engine can 'listen' for you and let you choose from the best ones according to requested criteria.

Smart seed generator



Resonance	100.0%	thr	100.0%	Fluttering	100.0%	thr	100.0%
Spectrum difference between L & R	100.0%	thr	100.0%	Max spectrum difference between L & R	100.0%	thr	100.0%
Level difference between L & R	100.0%	thr	100.0%	Max level difference between L & R	100.0%	thr	100.0%
Maximize echo density	100.0%	thr	100.0%	Echo density buildup time	100.0%	thr	100.0%
Echo density fluttering	100.0%	thr	100.0%	Echo density difference between L & R	100.0%	thr	100.0%
Echo count similarity between L & R	100.0%	thr	100.0%	Echo similarity between L & R	100.0%	thr	100.0%
Panorama off-center	100.0%	thr	100.0%	Panorama fluctuation	100.0%	thr	100.0%
Width non-optimal	100.0%	thr	100.0%	Maximimize attack time	100.0%	thr	100.0%
Maximum echo difference between L & R	100.0%	thr	100.0%	Length	100.0%	thr	100.0%
Length difference between L & R	100.0%	thr	100.0%				

Set all thresholds 100.0%

Smart seed generator is an extremely powerful statistical tool that makes the plugin try several different seeds, analyse them and extract several perceptual factors, based on which it can decide whether each seed is good or not. It then sorts them based on these factors. You can browse the seeds simply by selecting them in the listbox containing the results. The best seeds are at the top of the list.

The generator is parallelized by default, which lets it exploit the computational capabilities of your computer and analyzes the LR module separately. When the parallelization is disabled, it analyzes the global output of the plugin by sampling its impulse response. This can be used to combine multiple LR modules for instance.

Stereo field, the biggest challenge in audio ever?

One of the reasons we are employing reverbs in the first place is to create a virtual stereo field with depth and space, for audio materials that are otherwise clean or boring or both. While creating depth is simple enough, just providing reasonable evolving echo density and stereo field is a completely different story and currently nearly all reverbs out there perform very badly in this respect. The main problem is that the psychoacoustics here are extremely complex. Since birth our brains learn to analyze different factors of what we hear and associate them with what we see so that we can feel the place even when we do not actually see it at that moment. We can also detect where the sound originates from, potential obstacles close to us etc.

The problem here is that algorithmic reverbs do not care about that at all, but they still generate these clues, which often cause our brains to be confused. But brains are amazing devices and they adjust so quickly that, if you are designing a reverb, they actually work against you, fooling you into believing that your reverb actually "sounds good". And it just might sound good on its own, but in the context of a mix, or other tracks it may be actually quite bad. And pretty much all reverbs on the market that we have tried are.

What are the symptoms of a problematic stereo field?

Well, first of all, to spot the problems, you will have to train your ears. And it's not going to be easy. In fact, it's probably the hardest thing I have ever tried to teach my ears, well, my brain. But after you do that, you will be changed, as I am right now :).

First put your headphones on, get a monophonic recording that you are used to. Just a short loop is all that is needed. Always use the same loop, because sooner or later you'll know everything about it. This loop should contain both drums (being a source of noise-like signals) and melodic instruments (being a source of signals with harmonic relationships), but both should be percussive, so that you can always "feel" the reverb tail. Enable the reverb, ideally with dry/wet at 100% wet, but audition only the "mid" signal (mono). To do that you can for example add MStereoScope and use the mid/side selector. Or even better use MCompare, which has a **Mid** switch, since switching this way is much easier, and it is useful for comparing different reverbs anyway. Now, listen to the loop for as long as you need, until you feel that your ears hear the same thing spectrally each time and you feel as if everything is in front of you, far away, in some sort of tunnel. That's how mono sounds.

Now disable the Mid switch, and the sound will expand to stereo. Please note that you must NOT allow any pause in-between, you need a sudden change from mono to stereo. How you do that depends on the capabilities of your DAW. Listen for a few seconds and switch back to Mid. Try that with a few different reverbs, just to learn the differences. After a few attempts your ears should have some knowledge of the process, and now when you switch to stereo, a few things may happen:

A) Everything is fine, the sound expands around you, both ears seem to hear the full spectrum of frequencies, all instruments sound in both ears, and when you disable the reverb, it nicely decays to the center in front of you.

B) Something is wrong, and right now you are probably just confused and not sure what it is. Some classic problems that you may encounter are:

- The sound feels as if it is coming from one side. This is usually caused by the first echoes coming in one channel sooner than in the other. The delays need to be different, otherwise it would all sound mono again, but if there are many echoes coming from say the left channel ahead of the right channel, then it would seem as if the audio originated on the left. Or there may actually be more echoes on the left altogether. And there are plenty of other things that may go wrong.

- It seems like something is missing in one channel, as if one of your ears suddenly became partially deaf. The echoes together change the spectrum in some way and in many cases they cause an attenuation of an interval in the spectrum, just like an equalizer. The good thing is that brain will quickly compensate, because it just needs you to hear properly, that's a necessary survival skill. But it's actually a bad thing, because after a few seconds you feel as if everything is fine, so you need to switch to mono and let your ears adjust again.

- It seems that the bass drum (or another instrument) originates on the left. This is actually sort of a combination of both the previous cases. Your brain just really tries to make sense of all that, so it can actually group the frequencies that predominate in each channel into the instruments it can find there, and make you feel as if the bass drum is slightly on the left, the snare is on the right etc. The reverb basically causes fluctuations of the spectrum in time; that's what echoes do when they are close to each other. And in our case it means that in the beginning of the reverb tail most of the bass drum frequencies are present in the left channel and vice versa, so it seems like the bass drum is on the left. It's especially noticeable in the beginning of the reverb tail, where the echo density is quite low.

- After you stop the playback, the reverb tail decays, somewhere, but definitely not in the center. There are many things the reverb tail can do depending on the algorithm. Sometimes it goes from left to right and back again like an autopan. You might have even used an autopan, which would explain a lot :). But in most cases it just feels like the sound is running away to the left or right. Just a little. But if you are aiming for natural space reverberation, the sound should ideally stay as close to the center as possible. But again, to judge the center, you need to hit the mono switch, otherwise you just cannot be sure the brain is not deceiving you.

These things may actually be wanted for some creative purposes in some cases, but not in general. When mixing with such reverbs, you may get fooled and at one point you may just start asking yourself what is wrong, but in the context of multitrack mixing it will be pretty much impossible to detect that the reverb is the problem and why. You can always pan a nicely balanced reverb, but if its stereo field is muddled in the first place, as it is with most reverbs, there's just nothing you can do afterwards. And the difference is unbelievable, once you understand what to listen to.

It's also worth noting that the **Room type** control that most devices have is great for detecting these stereo field irregularities, because when quickly changing the settings, it's quite easy to perceive that the audio is jumping from one place to another in the stereo field. While it is just fine if the audio is changing distance - close and far - it's not so good when it is jumping from left to right and vice versa. This method works very well in detecting which settings are not ideal.

Ear fatigue

It's possible that after say 30 minutes you will feel as if it doesn't make sense at all anymore. What seemed fine to you sounds really bad now, and in a few minutes it may all change, so you cannot trust yourself anymore. That's normal. Your brain has been overloaded. It's as if you are teleporting from one place to another every few seconds and your brain needs to adjust all the time and at some point it just got tired. So take some rest, do something completely different and try again in an hour or so. After all, this is a good practice for all audio processing jobs. Our brain just likes to trick us; in this case it is just a little extreme.

How to balance the stereo field

The answer is the seed. It defines the delays, coefficients, everything under the hood, which is hidden from you as it is too technical and would be way too hard and clumsy to manipulate directly. But there are more than 2 billion seeds, so going through all of them to find the best one isn't really an option. The smart seed generator can cover thousands of seeds in just a few minutes and cannot be fooled that easily. On the other hand its ability to detect these psychoacoustic clues is highly limited, so at the end of the day your ears will have to be the final judge anyway. But the generator can still remove the seeds which have too many resonances, low echo density, or which don't seem to be centered or wide enough, saving a lot of drudgery.

The usual problem is that these factors often work against each other. For example, the higher the echo density, the higher the probability of resonances. That's why the smart seed generator may look a little scary. It lets you choose those factors that you want to be considered more important.

So to make your life easier, just start like this: First of all, set up your algorithm properly, including the settings in the **Designer** panel. Then open the smart seed generator and let it start searching for seeds. Give it at least 1000. The more the better, sometimes 100 is enough, sometimes not, depending on the algorithm. Stop the search, click the **Set 100%** button to make all factors equally important, then click the **Calibrate** button to let the engine make some statistical adjustments. And finally use the **Set all thresholds** value to set it as low as needed so all that os left is just a few dozen seeds. This will basically filter out the bad seeds, those which do not qualify in some of the factors. But do not set it too low, it's a computer, not an artificial intelligence; it may not know what you really want. And then try different seeds. Just click on each while listening. And when you feel that it is great, do not forget to do the "mono test", you may be very unpleasantly surprised.

Stereo width

One particular problem is that we expect reverbs to sound wide. But the width is caused by differences between the left and right channels, whether these are caused by different spectra, echoes or something else. And settings with a non-centered stereo field may also appear wider, initially. So do not be fooled by the impression of width. After all, the widest signal ever is out-of-phase and that's what we try to avoid at all costs. And as demonstrated by MStereoSpread, it is actually possible to have a stereo field that is centered, yet extremely wide. A good seed can provide great width and still a balanced stereo field, but it may be a little hard to find one.

Presets

Presets

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

Left arrow

Left arrow button loads the previous preset.

Right arrow

Right arrow button loads the next preset.

Randomize

Randomize button loads a random preset.

Copy

Copy button copies the settings onto the system clipboard.

Paste

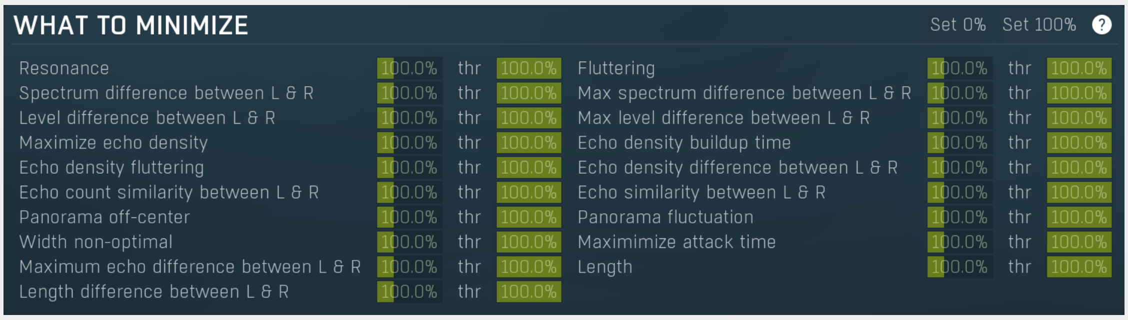
Paste button loads the settings from the system clipboard.

Random

Random

Random button generates random settings using the existing presets.

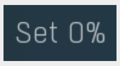
What to minimize panel


WHAT TO MINIMIZE — Set 0% Set 100% ?

Resonance	100.0%	thr	100.0%	Fluttering	100.0%	thr	100.0%
Spectrum difference between L & R	100.0%	thr	100.0%	Max spectrum difference between L & R	100.0%	thr	100.0%
Level difference between L & R	100.0%	thr	100.0%	Max level difference between L & R	100.0%	thr	100.0%
Maximize echo density	100.0%	thr	100.0%	Echo density buildup time	100.0%	thr	100.0%
Echo density fluttering	100.0%	thr	100.0%	Echo density difference between L & R	100.0%	thr	100.0%
Echo count similarity between L & R	100.0%	thr	100.0%	Echo similarity between L & R	100.0%	thr	100.0%
Panorama off-center	100.0%	thr	100.0%	Panorama fluctuation	100.0%	thr	100.0%
Width non-optimal	100.0%	thr	100.0%	Maximimize attack time	100.0%	thr	100.0%
Maximum echo difference between L & R	100.0%	thr	100.0%	Length	100.0%	thr	100.0%
Length difference between L & R	100.0%	thr	100.0%				

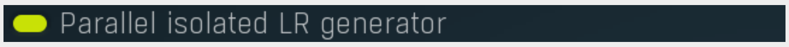
What to minimize panel controls which factors the seed generator should focus on. Each factor has a weight, which defines how important it is. If you set all weights to 0%, the list of results won't be sorted at all. If you set all weights to the same number (probably 100%), each factor will be considered equally important. Whenever you change any of the weights, the list of results gets sorted again to provide the correct order.
All factors are defined to be ideally as small as possible; hence the generator tries to minimize all values. For instance **Resonance** factor defines how much the IR seems to be resonating at some frequencies. Resonating usually sounds metallic and is rarely considered a good quality of a reverb, hence if you want focus on seeds that have minimum resonance, simply increase the weight for this factor.
Note that most factors are perceptual and as such as it is extremely hard (if possible at all) to quantify them well, as that

means simulating human auditory system. The generator performs a complex analysis of the impulse responses produced by the plugin, but the quality of the approximations may still vary a lot. Also note that since the analysis is based on impulse responses, it doesn't notice any kind of modulation, which varies in time (that is variations over time).
Next to each factor there is also a threshold slider, named **thr**. This lets you filter out results, that don't fit certain parameter at all. This lets you solve more complex situations, such as when you want to minimize all factors at the same time, but you do not want any resonance at all. Such a wish wouldn't be possible with just the weights, because that would make you increase the weight for the resonance to maximum, so the list would be ordered mostly by resonance, nearly ignoring other parameters. Please note that in order to make this work well, calibration needs to be performed. The threshold represents the maximum allowed value for the factor.



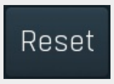
Set 0%
Set 0% button sets 0% weight for all factors, which effectively disables any sorting. It is useful when you want to focus on just a few factors and disabling all of them manually would be tedious.

Set 100%
Set 100% button sets 100% weight for all factors, which effectively enables them all and makes them equally important, which is default.

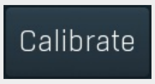


Parallel isolated LR generator
Parallel isolated LR generator button activates a special processing mode which uses multiple cores of your CPU and isolates the processing to only the single LR generator, ignoring all other modules and advanced options. In fact it only cares about the settings in the **Designer** panel. As such it cannot be used to combine multiple LR modules together for example, but it is several times faster and it lets you get ideal settings for individual LR modules.

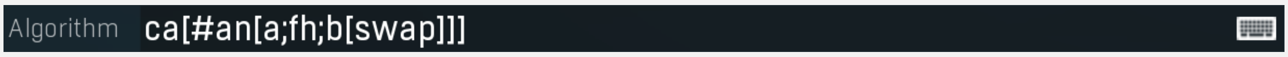
Perform smart seed search
Perform smart seed search button starts or stops the processing. During the processing the plugin cannot be used as the sampling requires exclusive access to the processing engine. You can stop the processing and continue at any time. The generator automatically detects the reverberation length and the longer it is, the slower the analysis works.



Reset
Reset button deletes all results.



Results
Results contains the list of tested seeds. The list is sorted by the first number, which is the total 'quality', which depends on the factors and the weights you selected for them. The lower this value is, the better. But always note that scientific measurements may produce quite different results than human perception. The quality value is followed by the actual seed and all the factors.


Algorithm ca[#an[a;fh;b[swap]]]

Algorithm
Algorithm controls how the Late Reflections generator actually works and you can program it completely using a single formula exploiting the many modules that the software provides. If you have reached this stage, it probably means that you are ready to get seriously creative and this is exactly what it is all about.

For decades scientists have been trying various "reverb topologies" to mimic the sound of real rooms (or just to get creative) without convolution, which is extremely CPU demanding and very limited when it comes to modulation and adjustment. **Topology** is, simply put, a set of basic building blocks connected in some way, each of which carries out a particular operation. Most reverbs on the market are usually a product of one such topology. When designing MTurboReverb we originally wanted to include just one single topology like everyone else, but as we were comparing it to different reverbs on the market, it turned out that a far bigger picture is actually possible. So we came up with the (almost) ultimate reverb modular system, which is not only particularly versatile, but also easy and quick to use, and extremely CPU efficient.

Instead of providing a complex graphical interface to define a topology, the algorithm is entirely constructed from a single line expression. The coefficients and delay lengths are some sort of pseudorandom numbers generated automatically (via the **Seed** and other parameters in this panel). The coefficients are order, focus and width for each of delay, input and output and are displayed in the Designer panel. Their usage depends on the algorithms. They are mainly relevant for the R, RD, CC and CA modules.

The algorithm only controls what modules are used and how they are connected (that is, it defines the topology) and the parameters are generated making it simple to set up the whole thing. The engine knows how many parameters and delays are used and how quickly the reverb should decay (please note though that a proper compensation is not always possible, especially with nested modules, see below). All modules are designed such that they compensate for gain change etc., so the hard work has already been done, and now the only thing for you to do is to experiment, use your ears, and be creative.

Introduction

Before you start programming your own reverberation algorithms it is highly recommended to check out the presets. You will learn a lot from a close inspection of them. Also you should enable the safety limiter button (on the right-hand side of the plugin), since infinite feedback is quite common when dealing with complex feedback systems. In most cases the engine will compensate for everything, but sometimes it just cannot be done. If you are ready for some programming, here's how it works. First a few examples, so that you can see how easy it actually is:

parallel[comb;comb;comb;comb];allpass;allpass = 4 combs in parallel followed by 2 allpasses in series. Those two modules types, and many others, are described below. This is actually one of the first reverbs ever designed, called Schroeder's reverb. It has many variations, but this is pretty much the basic one.

p[4c];2a = this is the same thing! It's just presented in a more compact syntax, it all depends on which you prefer.

reverb or *r* is a full reverb implementation. It's actually the topology that we originally wanted to include as the only one.

Syntax

The syntax defining the topology consists of several modules, placed one by one and separated by a semicolon ';'. Some modules can have parameters, placed in round brackets '(parameter1;parameter2;...)', and some may have submodules, placed in square brackets '[submodule1;submodule2;...]'.

In front of each module you can specify the number of occurrences, 100 is the maximum for safety reasons. The number only serves as a shortcut, so that you don't need to specify the same module multiple times, for example 3a is exactly the same thing as a;a;a. Instead of a number you can use a hashtag '#', which will be replaced by the value of the **Complexity** parameter. This way the user can change the topology without editing the algorithm formula itself at all, by changing the **Complexity** value.

You can insert spaces anywhere; these have no meaning, but they can make the algorithm easier to look at. You can also mix uppercase and lowercase letters. Most modules have a short name and a long name, it doesn't matter which you use.

For some modules there are parameters that you can either specify or they will be randomized. By randomizing we mean a controlled pseudo-randomizing based on the seed value, which means that it will be the same every time, until you click **Seed** again. The Seed is just a very long number, from which the whole pseudo-random sequence of numbers starts. The seed is stored in the same way as any other parameter, it can be attached to a multiparameter etc., so there's no black magic, it's just a very long ugly number :). Please note that whenever we talk about random numbers below, we actually mean these seed-based pseudo-random numbers.

To summarise, the syntax looks like this (the elements in italics are optional):
count/# **module** *(parameter1;parameter2;...) [submodule1;submodule2;...] ;*

User parameters and expressions

Most modules are designed in such a way that they don't need any further control. They calculate all the internal variables to maintain the desired reverb length, compensate for output level changes etc. But in some specific cases you may need to change the algorithm in some way, without actually accessing the text.

For that, the plugin provides 3 user parameters below the algorithm field, named **Param 1**, **Param 2** and **Param 3**. You can then implant those parameters anywhere you want in the algorithm by writing **$1**, **$2** or **$3**. These values are inserted as text, which means that the algorithm must make sense after replacing the $1, $2 and $3 literals by the values. In most cases you'd use these to alter the feedback coefficients, levels etc., where the available range of -1 to +1 is sufficient.

It is also possible to insert mathematical expressions featuring the parameters. These need to be delimited by **{expression}**. The expression evaluator will process the parameters, numbers and most of the useful mathematical functions. As an example, you can write *{$1 * 10 + sin($2)}* in the algorithm, whatever the purpose of such equation would be. Note that expression evaluation certainly requires a little more CPU when evaluating the algorithm, however there is no difference when it comes to actual audio processing afterwards.

Need to know information about reverb design

Feel free to design your reverbs just using your ears. You can just take some presets and use the randomizers to change how it sounds. But if you are willing to dig further, there is some more information you might need to know.

Frequency response is one of the big concerns. For example, try this as the algorithm: **3c**. It's 3 comb filters in series and it sounds horrible (if you are aiming at realistic reverberation). The problem is in the metallic resonances - a very specific frequency response. Actually it's sort of possible to get such a response in nature, well, it is if you are willing to sit in a small metal cube :). In any case it's not exactly how great ambiences sound. It completely kills some frequencies, while resonating at others. The problem is that it cannot be completely avoided. But one of the characteristics of good reverbs is that they do not resonate much, in other words their frequency response is reasonably flat.

Room modes are sort of the same thing as the frequency response. Each mode is a resonance. A little experiment - go to your toilet/bathroom and clap with your hands. And just listen. What you will probably hear is almost an alien-like sound. Well, these are the resonances. Now go to your nearest church, and I mean a real old church, a Gothic one ideally. And do the same thing - people may look at you as if you are weird, just tell them that it's for science! I do that every time I visit a church, because it's awesome! Your clap will slowly decay to a slightly filtered white noise, in a very very natural way. It's almost addictive. And guess what, there are still modes / resonances! But there are so many of them that you cannot hear them anymore, they just sort of cancel each other out. And that's more or less what we want to achieve. Sadly, it's probably not possible for short ambiences, since each echo is like a single resonance, so we need lots of echoes. That's not always possible, but we want to get close.

Echo density sounds like a fancy mathematical term, but is actually pretty simple - it just says how many echoes there are per second. After all, reverberation is nothing else than producing lots and lots of echoes. There are many theories about what the echo density should be, but in general in the beginning there's just one "direct signal" and then the echo density should ideally increase in time. Imagine a simple cubic room. You make a directional sound (not possible I know, but just imagine it), it reflects from a wall in all directions, so from one wave of sound you have several, and that happens every time a sound wave hits a wall. So at the end the number of echoes grows exponentially in time. And that's a problem, because the basic elements used in reverb design don't change the echo density over time, so to build a realistic reverb, we'll need some more complex topologies.

So these are the scientific properties for reverb realism. That said, it's absolutely not true that a reverb has to sound realistic to sound good. Just look at plate reverbs, they are commonly used these days despite they are everything but realistic. There are no limits to creativity!

Helper modules

The first set of modules doesn't do anything directly to the sound, but these are necessary for managing other modules.

S / **Serial** [submodule1;submodule2;...]
Executes submodule1, submodule2 etc. in series, one by one. Since this is a basic module it is the default and you can omit its name. Thus (*[4c]* is the same as *s[4c]*).

Example: *[3c;2a]* = 3 combs and 2 allpasses in series, the first comb receives the dry signal and the other 4 receive the audio from the previous stage, so the output = comb1->comb2->comb3->allpass1->allpass2

P / **Parallel** [submodule1;submodule2;...]
Executes submodule1, submodule2 etc. in parallel and mixes their outputs together.

Example: *p[3c;2a]* = 3 combs and 2 allpasses in parallel, each receiving the dry audio signal, so the output = comb1+comb2+comb3+allpass1+allpass2

PW / **ParallelW** *(w1;w2...)* [submodule1;submodule2;...]
Executes submodule1, submodule2 etc. in parallel and mixes their outputs with the specified/randomized weights. A weight is just a mathematical term for a multiplier or level, so, for example, 0.5 halves the signal level, -1 inverts it.

Instead of writing the weights themselves, you can just enter "c", which stands for coefficients. In that case the module uses the input coefficients for the weights. It is not enabled by default, because we usually want a completely random sequence independent of the input coefficient system, which can process and sort them.

Example: *pw(1;-1;0.5)[3c;2a]* = 3 combs and 2 allpasses in parallel, where the weights for the first 3 combs are specified in the brackets and for the allpasses the weights are randomized.

Example: *pw(c)[#a]* = several allpasses in parallel, the number of allpasses depends on the **Complexity** parameter, and the weights are defined using input coefficients. This algorithm sounds more like an effect than a reverb, but it's behaviour depends a lot on the input coefficients and you can variously modify/sort them using the input coefficient parameters. That way you can for example make allpasses with longer delays more prominent.

B / **Bypass** *(probability)* [submodule1;submodule2;...]
Executes submodules serially, so it's similar to **serial**, but it may bypass the whole chain based on the probability in the range 0..1 (default 0.5, which means 50% probability). For example, 0.1 means 10% probability that the modules will be bypassed. This simple module is great for randomizing topologies and also for saving resources - some modules may be quite CPU hungry and you may not need all of them, so this is an elegant way to enable only some of them. If the bypass is active, all resources from the submodules are freed, so you don't need to worry about efficiency.

Example: *4b[a]* = 0-4 allpasses in series, how many? Depends on the seed ;)

Example: *b(0.1)[10a]* = 0-10 allpasses in series, but with a probability of bypass 10%, it is more likely that there will be 10 of them ;)

Example: *10b(0.1)[a]* = 0-10 allpasses in series, each with some random chance of being active, hence 10 bypass modules, each with one allpass.

L / **Level** *(gain in dB)*
Level module performs a simple gain. When there is no gain specified, it does nothing, literally. That way it can be used as a dummy module which is sometimes useful for more complex algorithms. Instead of gain in dB you can also specify one of the keywords 'in' or 'out', which lets the module allocate and use an input or output coefficient instead. This way you can perform a random level change and control the sequence using input or output coefficient parameters.

Example: *p[l;4c]* = a level module and 4 comb filters in parallel, each receiving the incoming signal itself, and mixed together.

Example: *l(-3)* = a simple -3dB gain.

Example: *p[l(-3);4c]* = a level module (reducing the level by 3dB) and 4 comb filters in parallel, each receiving the incoming signal itself, and mixed together.

Example: *p[#[c;l(in)]]* = parallel comb filters, each with a random gain depending on the input coefficients.

LC / **LevelC** *(gain as multiplier)*
LevelC does exactly the same job as Level, but its parameter is specified as a multiplier instead of a gain in decibels. So the module simply multiplies the input signal by the specified value. It is mainly useful in conjunction with the user parameters, so that you can for example tweak the feedback coefficients.

Example: *an[lc($1)]* = an allpass diffuser with a level module in its feedback path, controlled by the **Param 1** value, which can be used to lower the feedback level.

ST / **SerialTap** [submodule1;submodule2;...]
Executes submodules serially, so it's similar to **serial**, but the output is taken by mixing the outputs from each of them as opposed to normal serial processing which takes only the output of the last submodule.

Example: *st[#d]* = a simple tapped delay having # taps.

Example: *st[3a]* = 3 allpasses in series with their outputs mixed, so it's like the output = (a1) + (a1->a2) + (a1->a2->a3). While the serial tap has been used a lot for reverb design in the past, I personally feel it is quite obsolete, because of one danger - (a1->a2) is just a modified (a1), so there's a strong chance of comb filtering, leading to some sort of metallic result. So if you're heading in this direction, it will probably be useful to perform some kind of modulation in each stage.

STW / **SerialTapW** *(w1;w2...)* [submodule1;submodule2;...]
Similar to serialtap, but weights the outputs being mixed. You can, again, use "c" instead of the actual weights to make the engine use input coefficients.

Example: *stw(1;-1)[3a]* = 3 allpasses in series with their outputs mixed, where the weights for the first 2 of them are specified, the 3rd weight is randomized.

Example: *stw(c)[#d]* = a tapped delay having # taps (defined by the **Complexity** parameter), but the output from each tap is weighted using the input coefficients. By sorting the input coefficients into ascending order you can create a reverse delay for example - sorting the weights into ascending order makes each tap a little louder than the previous one.

Basic modules

These are the basic building blocks of all reverb design. All of them are based on delays, sometimes with a feedback, sometimes without.

D / **Delay** *(time)*
A simple delay, which shifts everything in time. On its own it doesn't provide any kind of reverberation, but it can be useful with other modules. The delay time is specified in milliseconds. If not specified, it's randomized.

Example: *p[c;[d;c]]* = 2 comb filters in parallel, but the second one is delayed by a random amount.

C / **Comb** *(decay;time)*
Comb filter, a delay with feedback, which is basically just an infinite echo. It is called a comb filter because its frequency response looks like a comb, literally. It produces resonances at multiples of the base frequency (depending on the delay length), hence if the base is say 100Hz, then there will be resonances at 100Hz, 200Hz, 300Hz etc. And that's not good, because it sounds metallic. It's like listening through a metal pipe. If the delay length is long enough, above say 40ms, then mathematically it's still a comb filter, but the brain recognizes each echo separately and you don't hear the metallic behaviour any more, but for reverberation one usually needs far shorter delays, so that's not a solution. However the metallic effect can be diminished to a large extent by placing comb filters in parallel, so that statistically their resonances will counteract each other.

Decay controls the decay coefficient, which is 1 by default, which means that the comb filter's decay follows the **Length** parameter - by definition the reverb length is the time period until the impulse response decays below -60dBFS. Using this parameter you can speed up the decay, or invert it using -1, or if you specify 0, the decay factor will be randomized. Time is the length of the comb filter's delay specified in milliseconds and, as always, randomized if it is not defined.

Example: *p[#c]* = comb filters in parallel, the number depends on the **Complexity** parameter. This can actually sound like a reverb, not a very good one though.

Example: *c(0.5;10)* = a comb filter with extra fast decay and delay of 10ms.

CB / **CombB** *(decay;time)*
This is an alternative to the **Comb** module, which lets the direct signal pass as well, unlike the previous comb, which produces only the echoes.

X / **XComb** *(decay;time)* and **XB** / **XCombB** *(decay;time)*
This is another alternative to the **Comb**, with a different kind of level compensation. The normal comb compensates for the gain in

such a way that the overall level stays the same. This is usually the best way to go, however since the comb filter changes the frequency response, some frequencies are attenuated and others are amplified. That means that if you place such a comb filter into a feedback path, the level of frequencies that are amplified would rise with each echo and there you have it - an infinite feedback.

xcomb compensates in a different way - it makes sure that no frequency gets amplified. As such it is lower in level, so it may be a little too silent for normal use, but it is the solution in those cases where you want to use it inside some feedback path.

A / **Allpass** *(decay;time)*
Allpass is a more complex comb filter, which doesn't alter the spectrum, mathematically speaking. In practice it highly limits the metallic resonances, but as it usually is, it's not perfect either. Its response is a bit unnatural, as the decay first builds up and then goes down in a way, and so it sounds good only for small delays. As such. allpasses are mainly used as diffusers; a comb filter (or something else) produces somewhat sparse echoes and an allpass then creates additional reflections in between - the diffusion. As such, allpass filters usually employ small delays, shorter decay and are used in series.

With higher decay coefficients allpasses can sound sort of metallic too and since the usual usage is as diffusers, we defined the default decay as 2/3 = 0.6666... If you want it to react similarly to **Comb** and have the decay according to the **Length** parameter, specify the decay parameter as 1 (or -1).

We call this module just allpass or allpass diffuser, to avoid confusion with allpass filters, which will be mentioned later too.

Example: *p[#c];3a* = classic reverb design featuring several comb filters in parallel followed by a diffusion section made by 3 allpasses in series.

Filtering and modulation

Most problems in reverberation algorithms are caused by the fact that the original signal being delayed and summed back is always the same. This behaviour itself is also very unnatural. Filtering and modulation modules are essential to get more realistic reverberation and dampening and to avoid the metallic resonances caused by summing these exact same signals.

V / **Vibrato** *(length;lfo frequency)*
Vibrato is a classic modulation processor, which varies the pitch coming through it, while also delaying it a little. In fact, the global modulation panel in the LR tab does exactly this, but on the input signal. The Vibrato module lets you perform this pitch variation anywhere and is especially useful for feedback processing, which will be discussed later.

Length controls the delay line size and can be specified in milliseconds, the default is 1ms and maximum is 10ms. LFO frequency controls how quickly the pitch is modulated, in Hz. If it is not specified, a random value is used. If both values are too high, the resulting effect can be very unnatural, kind of "honky tonky". If any of the values is too low, the effect may not be noticeable at all.

Example: *p[#[c;v]]* = parallel comb filters, each with a vibrato modulating its output. It's a basic, but powerful, technique to make the outputs of each comb filter different enough to minimize the metallic resonances.

Example: *p[#v]* = parallel vibratos. This is actually a simple chorus effect.

FL / **FilterL** *(frequency diff;Q diff)*
Filter low implements a low-shelf filter controlled by the **Dampening Low** panel. Without this module (or some complex module, which implements it internally) the dampening panel will not have any effect. By default it follows the settings in the panel strictly, however you can change the frequency and Q values randomly. Frequency diff controls the maximum number of octaves that the actual filter frequency may differ from the settings. Q diff controls the maximum difference in Q specified by a 2^x multiplier - if you set it to 0, which is the default, no randomization will take place. Set it to 1, and the actual Q may go from 50% (1/2x) to 200% (2x) compared with the settings in the **Dampening Low** panel.

Example: *p[#[c;fl(1;0.1)]]* = parallel comb filters, each with a low-shelf filter (if the dampening is enabled), where the the frequency of each filter may go from 1 octave below the value in the **Dampening Low** panel to 1 octave above, and Q may be a little different as well (about +/- 0.7%).

FH / **FilterH** *(frequency diff;Q diff)*
Filter high implements the high-shelf filter controlled by the **Dampening High** panel. It works in the same way as **FilterL**.

FA / **FilterA** *(frequency;Q)*
Filter allpass implements a second order allpass filter with its center and its Q value both either random or specified. It is important to understand the difference from the **Allpass** module, which is based on a longer delay line and should produce distinct echoes. FilterA has no delay (well, strictly, it has 2 single sample delays) and its purpose is not to produce any kind of echo, but to change the phase of some frequencies in the signal while keeping their levels intact. It provides a way to make a signal different from the original without actually causing very noticeable changes to it when played on its own. That way it is an alternative to the **Vibrato** module.

Example: *p[#[c;fa]]* = parallel comb filters, each with an allpass filter modulating its output. It's another powerful technique to make the outputs of each comb filter different enough to lower the metallic resonances.

FAM / **FilterAMod** *(modf;modq;modLFOFrequency;center frequency;center Q)*
Filter allpass modulated is an extension to the **FilterA** module, where the frequency and Q of the filter are modulated by an internal LFO. ModF then defines the modulation range of the filter frequency in octaves and defaults to 1, which means the actual filter frequency is oscillating 1 octave up and down around the center frequency (which may be specified or randomized). ModQ defines the modulation range of the filter Q as 2^x and defaults to 0, as described for the **FilterL** module. Hence if this is 1, the Q will go

from 0.5x to 2x around the center Q (which may be specified or randomized). ModLFOFrequency specifies the frequency of the LFO in Hz.

Example: *p[#[c;fam(0;1)]]* = parallel comb filters, each with an allpass filter modulating its output. The frequency of the each filter does not change (as modf is 0), but the Q is changing from half the random center to double that value.

F / **Filter** *(frequency;type;Q;gain)*
Filter lets you create a classic filter as used in most equalizers. You should specify all of the required parameters, otherwise these will default to frequency=8000Hz, Q=0.7071 (neutral Q), gain=-6dB and type=lowpass6. The filter type parameter is self-explanatory can have one of following values: Peak, PeakAnalog, LowShelf, HighShelf, LowPass, HighPass, LowPass6, HighPass6, BandPass, Notch (case-insensitive).

Example: *f(1000)* = a 6dB/octave lowpass filter at 1000Hz. 6dB/octave filters don't have a resonance nor gain, so the 2 parameters, which haven't been specified are ignored.

Example: *f(2000;highshelf;2;-8)* = a highshelf filter at 2kHz with Q=2 and gain -8dB.

Sat / **Saturation** *(drywet;mode)*
Saturation implements exactly the same algorithm as the one used in other plugins from MeldaProduction, e.g. MSaturator. You can control its dry/wet ratio or it will be randomized and mode sets the kind of the saturation, currently from 0 to 8 (defaulting to 0).

Please note that saturation is a property of analog components and it does not occur in natural reverberation systems, but it's a common way to phatten up the signal, that's the reason for providing this module.

And that since we assumed that this module could be used in feedback, we couldn't allow it to increase the level of the signal, otherwise an infinite feedback would occur. So instead it lowers the level, which would speed up the decay when used in feedback paths. It's not technically a bad thing, but it is important to be aware of that.

Example: *p[#[c;sat]]* = parallel comb filters, each with a saturation phattening its output. Please note that this will not help avoid metallic resonances very much, if at all.

Nested modules

Nesting is an essential method for designing reverbs with more complex feedback. So far we have described 2 modules featuring a feedback - **Comb** and **Allpass**. In both cases the feedback created from the delayed signal and the input signal is not processed in any way. Nesting lets you process the feedback line which is coming from the delay output back into the delay input.

There's an interesting problem with the nesting - the reverberation length. Currently the modules are able to actually find out what delay the submodules cause and to compensate for it appropriately. However note that for example putting an allpass diffuser inside a feedback loop of a comb filter will actually make its response longer, because the tails of both modules will interact and there's no way to compensate for such a complex behaviour. You can manually compensate using the **Level** module in the feedback path.

CN / **CombN** *(decay;time)* [submodule1;submodule2;...] and **CNB** / **CombNB** *(decay;time)* [submodule1;submodule2;...]
Nested comb filter is an extension to the **Comb** module, which lets you process the feedback path with specified submodules, executed in series.

Example: *cn[fh;fl]* = a comb filter with both dampening filters in the feedback path, which is a classic approach used to dampen the reverberation tail continuously.

Example: *cn[a]* = a comb filter with an allpass diffuser in its feedback path. Please note that such a processor may not follow the reverb **Length** any more.

XN / **XCombN** *(decay;time)* [submodule1;submodule2;...] and **XNB** / **XCombNB** *(decay;time)* [submodule1;submodule2;...]
This is an alternative to the **CombN**, with a level compensation that prevents any frequencies being amplified by the comb itself. Please refer to **XComb** for more information.

Example: *cn[xn[xn[fh;fl]]]* = a comb filter with another comb filter in its feedback path, which also has a comb filter in its path. The innermost comb filter contains dampening filters in its feedback path. The inner combs need to be XN and not CN, as CN keeps the level intact, but amplifies the resonance frequencies, so you would get an infinite feedback with multiple CNs.

CNI / **CombNI** *(decay;time)* [submodule1;submodule2;...] and **XNI** / **XCombNI** *(decay;time)* [submodule1;submodule2;...]
This is an extension to the **CN** and **XN** modules. It executes submodule1 on the delay output (being the output of this module). The remaining submodules are executed on the feedback path the same way as with CN/XN.

Example: *cni[[2a];fh;fl]* = a comb filter, where the output is processed with 2 allpass diffusers in series, and that is fed back through dampening filters.

AN / **AllpassN** *(decay;time)* [submodule1;submodule2;...]
Nested allpass diffuser is an extension to the **Allpass** module, which lets you process the feedback path with the specified submodules, executed in series. Unlike **CombN** it doesn't alter the levels of the frequencies in the signal, so it usually sounds less metallic and is more suitable for execution in series.

Example: *2an[2a;fh;fl]* = 2 allpass diffusers in series, each with 2 allpass diffusers and both dampening filters in the feedback path. It actually works as a nice simple reverb.

Example: *p[#an[a;fh;fl]]* = several parallel allpass diffusers, each with another allpass and dampening filters in the feedback path. It is already a nice reverb but it is not very dense.

STFB / **SerialTapFB** *(w1;w2...)* [submodule1;submodule2;...;submoduleN]
Serial tap with feedback is an extension to **SerialTapW**, which processes all submodules up to the last one summing them into the output, but then it uses the submoduleN to process the output from the previous submodule and sends it back into the input. This is like a complex extension to **CN** and **CNI** modules. Please note that since the outputs from the submodules are mixed, again there's the problem with comb filtering. If you specify only one submodule, there won't be any feedback processing and the module will actually work like **CNI**.

The weights will be constructed the same way as with SerialTapW module - either they will be randomized or you'll specify some, or you write "c", in which case input coefficients will be used.

Example: *stfb[4d;v]* = serial delay with 4 taps summed together and a vibrato in the feedback to minimize metallic sound caused by the feedback.

Example: *stfb(c)[#d;a]* = a tapped delay having # taps, but the output from each tap is weighted using the input coefficients. The feedback is processed using an allpass diffuser. By sorting the input coefficients up you can create a reverse delay for example - sorting the weights up makes each tap a little louder than the previous one.

Complex modules

There have been decades of investigation into the field of artificial reverberation with the twin goals of providing good echo density and avoiding the metallic resonances. The Following modules pretty much cover the latest inventions.

CC / **CircularComb** [outmodule1;...;outmoduleN;fbmodule1;..;fbmoduleN]
You can view a circular comb as a set of comb filters that are executed in parallel, where the feedback from the 1st comb goes to the 2nd comb, feedback from the 2nd comb goes to the 3rd comb etc. This can successfully remove the metallic resonances, because there are no short equally-spaced echoes as in the traditional comb filter. The complex feedback system makes the signal "travel" through the whole system until it circulates back to the beginning, which is usually late enough so that the brain no longer hears equally spaced echoes.

This module will probably form a basis for most of your realistic reverberation experiments, so it is essential to know how to use it. It has no parameters, and for now it is the first module that makes use of both the input & output coefficients and delay lines, so it exploits the whole randomizing system of MTurboReverb. It expects an even number of submodules, 2 times N of them. The First N modules are processing the output of each delay, the last N modules are processing the feedback lines. The number of submodules therefore determine the number of the delays in the system, which is N. So with say 16 submodules, there will be 8 delay lines. If there were say 17 submodules, then the last one is "alone" and hence will be ignored and the number of the delay lines will still be 8.

Example: *cc[#l;#[fl;fh]]* = comb filter system where the number of delays equals the **Complexity** value and if you make it high enough, it actually sounds like a nice reverb. Level (L) modules are used as dummy modules doing nothing, hence the output from the delays is not processed at all. The feedback is processed using the dampening filters though.

Example: *cc[v;#a;#[fl;fh];fa]* = a more complex reverb, where the number of delay lines is **Complexity** + 1. It is + 1 as the very first module, a vibrato, and the very last, an allpass filter, are always present - the number of modules is 1 + # + # + 1, and the number of lines is (1 + # + # + 1)/2. The output of each delay line is processed using an allpass diffuser (to increase the echo density over time), except for the first one, which is modulated using a vibrato. The feedback lines are processed using the dampening filters, except for the last one, which uses an allpass filter, just for some modulation.

Example: *cc[#b[a];#b[a];#b[fl;fh];#b[fl;fh]]* = an even more complex reverb, where the number of delay lines is 2 * **Complexity** - the number of modules is #+#+#+#, and the number of lines is (#+#+#+#)/2. The output of some of the delay lines is processed using an allpass diffuser, and the feedback lines of some of them are processed using the dampening filters. This example nicely show the advantages of the **Bypass** module, which lets you randomly process each of the delay lines.

XC / **XCircularComb** [outmodule1;...;outmoduleN;fbmodule1;..;fbmoduleN]
Circular comb with compensated feedback, so that no frequency is amplified. This is needed when placing the circular comb inside a feedback path, because a **CC** module would most likely cause an infinite feedback.

CA / **XCircularAllPass** [outmodule1;...;outmoduleN;fbmodule1;..;fbmoduleN]
Circular allpass is similar to circular comb, but instead of comb filters it consists of several allpass diffusers with interconnected feedback lines. As such it doesn't produce an actual flat response even from a mathematical point of view. However in many cases it can indeed sound far less metallic than circular comb.

FDN4 and **FDN4D**
4x4 feedback delay network (FDN) of 2 types. Imagine 4 comb filters in parallel, where the feedback lines are interconnected, so that the output of each comb is actually sent to the other 3 comb filter feedback lines. FDNs provide a nice way to increase the echo density and diffuse the output, because while the echo density of a single comb filter is constant, it just creates evenly-spaced echoes, multiple comb filters affecting each other make the echo density rise in time, which is natural for physical spaces.

FDNs provide a mathematical apparatus to generalize other reverb topologies, but their design is very complex and they are rarely very effective, especially for the larger sizes needed to implement realistic reverberation. Therefore they are used mainly for theoretical research. We include 2 types of 4x4 FDN (FDN4D provides higher diffusion), which are useful along with other modules.

Example: *p[#fdn4]* = several 4x4 FDNs executed in parallel. You can view this as an extension to parallel comb filters, where FDNs provide more natural echo density, however there is still the problem with metallic resonances, hence the need to execute them in parallel.

XFDN4 and **XFDN4D**
Similarly to Comb and XComb, these are special versions of FDN4 and FDN4D, which compensate the level in such a way that no frequency gets amplified. Hence these should be safe to use inside feedback loops.

FDN *(size)* [submodule1;submodule2...submoduleN]
Generalized feedback delay network of specified size or controlled by **Complexity**. It is a complex version of FDN4 with size ranging from 1 to 256. In theory an appropriately-sized FDN can do the job of any other topology, unfortunately designing a proper matrix for that task could be extremely complex and the CPU consumption of such a module may be huge. This module implements a generalized class of matrices, which can provide a huge variety of ambiences, however it is imperative to watch the CPU meter when utilizing it. Since there is hypothetically an infinite number of matrices that the module can produce and these are one of the main factors controlling the output sound, you may need to try several **Seeds** to get the sound that you are searching for.

You may specify submodules to process the feedback lines. If you specify too many of them, the remaining ones will be ignored.

Example: *fdn[#[fl;fh]]* = a fully featured FDN based reverb of size # (that is, determined by the **Complexity** value), where each feedback is processed using the dampening filters.

XFDN
Similarly to FDN4 and XFDN4, this is a special version of FDN, which compensates the level in such a way, that no frequency gets amplified. Hence this should be safe to use inside feedback loops.

Example: *fdn[#[xfdn;fh;fl]]* = FDN, where each feedback line is processed with another FDN and dampening filters. This is one of the structures nobody normally even dared to explore as it is immensely complex, so its creative potential hasn't really been discovered yet.

R / **Reverb** and **RD** / **ReverbD**
Reverb modules are similar to **CircularComb**, but instead of single comb filters they use 4x4 FDNs, so they are even more complex. The ReverbD version provides higher diffusion. These are actually the modules that were originally intended to be the only 2 topologies in MTurboReverb. They have no parameters or submodules and everything is completely randomized. They exploit full randomizer potential, follow the **Complexity** parameter (the actual number of delay lines is 4 * complexity) and implement the dampening filters internally including the **Sparse** parameter.

Example: *r* = yes, that's the only thing you need to write to make a good reverb in MTurboReverb :).

RP / **ReverbP** [submodule1;submodule2...] and **RPD** / **ReverbPD** [submodule1;submodule2...]
Reverb with processing extends the **Reverb** modules by allowing you to process the feedback lines in some way. Please note that the number of delay lines is 4 * complexity, hence there are also that many feedback lines. The modules make use of as many submodules as they can. For example, if the complexity is 4, there will be 16 delay lines and 16 feedback lines. In this case you can therefore specify up to 16 submodules. If you specify more, the remaining ones will be ignored. If you specify less, some of the feedback lines won't be processed.

Example: *rp[#b[v(0.5)];#b[v(0.3)]]* = a reverb with 4 * **Complexity** delay lines, where the feedback of the first half may or may not be modulated using a vibrato with slightly delayed delay lines (just not to over-process the signals, since there will probably be many vibratos).

CROSS / **Crossover** *(slope;frequency1;frequency2..)* [band1;band2...]
Crossover module splits the input signal into multiple bands (depending on the number of submodules you specify) and makes the first one process band 1, The 2nd one process band 2 etc. After that it mixes everything back to generate the final output. The maximum number of bands is 32; any further submodules would be ignored.

By default the bands are uniformly split through the spectrum from 20Hz to 20kHz, but you can specify the band split frequencies (in Hz) manually. Slope parameter controls the crossover separation between bands as dB per octave and can be 6, 12, 24 (default), 48, 72, 96 or 120. If you specify a different number, the nearest lower number (or the smallest if below the minimum) is chosen.

Example: *cross(48)[l(-30);r;[5a]]* = splits the spectrum into 3 bands using 48dB/octave crossover. The lowest band (bass) is simply attenuated by 30dB, mid band is processed using a fully featured reverb, and the highest band (treble) is processed with 5 allpasses in series. Such a construction doesn't really make much sense, so it is for demonstration/creative purposes only.

Example: *cross[8v]* = 8-band vibrato.

Example: *cross[#[p[#c]]]* = splits the input into # bands and in each of them # parallel comb filters are processed in parallel. This actually sounds sort of like a spring reverb, depending on the parameters.

CROSSLP / **CrossoverLP** *(slope;tapcount;frequency1;frequency2..)* [band1;band2...]
Crossover LP module is similar for **Crossover** module, however it uses a linear-phase crossover. Please note that as such it introduces latency which can NOT be compensated within the plugin. Tapcount parameter controls this latency in samples (defined for 44,100 Hz sampling rate, it is adjusted for other sampling rates). The higher it is, the more accurate the crossover is, but the more delayed the output is. The default value is 512, which actually isn't a very high accuracy, but it is an optimal quality/latency ratio for 24dB/octave crossovers. You can specify values from 64 to 8,192.

Multi-channel modules

Most reverbs out there work in single-channel mode. That means that the input channels are merged in some way according to the **Cross** parameter (many reverbs actually don't have even that and simply mix all channels to mono) and after that they are processed independently. Usually that's just fine and speeds up processing, however some topologies require the channels to react to each other. It's especially noticeable if you feed the reverb with a very wide (or fully panned to either side) audio signal.

Please note that if you use any of the following modules, MTurboReverb will switch into multi-channel mode, which may consume additional CPU power. Also, the following modules do nothing when used to process monophonic audio.

Swap
Swaps the left and right channels. If used in surround mode, it swaps channels 1 & 2, then 3 & 4 etc. Swapping is the most used interchannel processing; as -it doesn't mix anything it lowers the risk of metallic resonances and it provides maximum interchannel width.

Example: *cc[#l;l;#[fh;fl];swap]* = simple, but nice reverb featuring #+1 interconnected delay lines with dampening filters in the feedback paths of all of them except for the last one, which swaps the channels creating a nice spatial effect for stereo audio materials.

Mono
Mixes all input channels to mono. It is the opposite of **Swap** as it narrows the stereo field. It is not useful too often, but it can be used to simulate narrow spaces and to make the stereo image more controlled.

Widening *(depth)*
Performs the classic mid/side width processing to control the stereo width. Depth is either randomized or can be specified, in the range -1 to +4. -1 means collapse to mono (but it is more effective to use the **Mono** module directly), 0 means no width processing, +1 means 2x bigger stereo width. Please note that even though the module compensates for level changes, placing it inside a feedback loop increases the likelihood of infinite feedback.

LRtoMS and **MStoLR**
LRtoMS converts the input signal to mid/side format and MStoLR converts it back to the traditional left/right format. In order for this to work properly you should always use them in pairs, like this: LRtoMS;{some processing};MStoLR. This can be useful to control the stereo field more accurately.

Example: *lrtoms;r;mstolr* = reverb processed in mid/side format. Please note that since most signals are pretty thin (that is, have a low width), monophonic in many cases, the Mid channel is usually far more dominant, hence processing in Mid/Side mode often leads to lower the output stereo width.

Example: *p[r;[lrtoms;r;mstolr]]* = 2 reverbs in parallel, one processing the input directly, the other processing it in mid/side format. It's like a compromise between the direct reverberation, which often leads to the stereo field being aimed slightly to the left or right side, and the mid/side being too thin.

Advanced modules

We decided to include more advanced modulation and helper algorithms, which you usually use as normal effects for processing your mixes. These turn out to be very effective for more creative purposes.

Tremolo *(depth;frequency)*
Tremolo is a classic volume modulation effect and is mainly useful for creative effects. Depth controls the dry/wet ratio from 0 to 1. Frequency is specified in Hz. Both parameters are randomized if not specified.

Autopan *(depth;frequency)*
Automatic panner can be helpful in providing additional stereo width and is mainly useful for creative effects. Depth controls the dry/wet ratio from 0 to 1. Frequency is specified in Hz. Both parameters are randomized if not specified.

Pitch *(fftsize;pitch shift;formant shift;keep formants)*
Pitch shifter is a powerful, but CPU- consuming, module, hence you need to be careful when using it. It also introduces latency, which cannot be compensated within the plugin, so technically it causes an additional delay. It can be used for various creative effects, usually in feedback paths.
fftsize controls the size of the processing buffer, hence also the latency. The higher it is, the higher the frequency resolution will be, but the lower the time resolution will be. So it's always a compromise. This value needs to be a power of 2 from 64 to 16,384, default value is 1,024, and the ideal values usually are 1024, 2048 or 4096. Please note that these values correspond to sampling rates around 48kHz and are recalculated for higher sampling rates. If you specify a value which is not a power of 2, it will be rounded up to the nearest higher power of 2.
Pitch shift value follows the global **Pitch shift** parameter or you can specify it, in semitones, in the range -12 to +12. You may specify value 'c', which makes it follow the pitch shift parameter and may get useful if you want to specify any parameters after this one, but do not want to specify the actual pitch shift. Formant shift is randomized or you can specify it in semitones as well, in the range -12 to +12. Keep formants must be in the range 0 to 1 and controls how much the formants will be kept intact. It defaults to 1 (that is, keep the formants fully intact).

Example: *cc[#a;#l;l;pitch(1024;-1)]* = a powerful circular comb system, which has a diffuser in the output of each comb filter to gradually increase the echo density, but the last one has a pitch shifter in the feedback path instead. Since the pitch shifter has a latency, the last comb filter uses a longer delay than it actually should, so it sounds like the pitch shifting is in some global feedback. It is shifting the pitch down by 1 semitone. For more typical "shimmer" effect use -12 or +12, meaning one octave up or down.

FreqShift *(maximum;shift;override)*
Frequency shifter shifts all frequencies up/down by the specified number of Hz, ranging from -10,000Hz to +10,000Hz. If the shift is not specified, the value is randomized in the range -maximum to +maximum, where the maximum defaults to 1000.
It's similar to pitch shifting, but it doesn't keep the harmonic relationship, so the higher the shift, the less natural is the output sound. With a very small shift, say below 10Hz, it can sound similar to pitch shifting, so it can be used like a less CPU-consuming method without latency. For higher shift values it can be used for creative purposes. Since it can modify the signal without changing its character too much, it is an efficient way to fight the metallic resonances.
Override parameter lets you use the $1 parameters. For example, *freqshift(100;0;$1)* makes the engine use parameter 1 to get shift from -100Hz to +100Hz.

Example: *cc[#b[freqshift(100)];#a]* = a very creative circular comb system, which gradually increases the echo density using the allpass diffusers, but also performs random frequency shifting at random places, hence it can provide all sorts of effects from quite natural reverberation to extremely creative stuff. Change the shift value of 100 to something higher to make the effect more creative.

Noise
Produces white noise. Useful for extremely creative ideas.

Mul *(drywet)* [submodule1;submodule2...]
Processes the input signal using the submodules and then multiplies the results by the input. Drywet parameter (in range 0..1, 1 by default) controls how much of the dry input signal is mixed with this to form the final output.
Example: *mul[noise]* = multiplies the input by the white noise level.

Spectral *(fftsize;attack;hold)*
Experimental spectral domain reverb, which is by no means realistic, but can sound nicely ethereal. It follows the reverb length and dampening automatically.
fftsize controls the size of the processing buffer, hence also the latency. The higher it is, the higher the frequency resolution will be, but the lower the time resolution will be. So it's always a compromise. This value needs to be a power of 2 from 64 to 16,384, default value is 1,024, and the ideal values usually are 1024, 2048 or 4096. Please note that these values correspond to sampling rates around 48kHz and are recalculated for higher sampling rates. If you specify a value which is not a power of 2, it will be rounded up to the nearest higher power of 2.
Both attack and hold values are specified in 10 seconds units, which is a little nonstandard, but it is done this way so that you can use the parameters, which have range -1..+1, well with them. Hence you can easily parametrize up to 10 seconds attack & hold, which seems to be a reasonable maximum. Attack defaults to 0.1 seconds (hence 0.01 if you use the parameter) and hold defaults to 0.

Example: *spectral(4096;$1;0.1)* = spectral reverb with higher resolution, attack controlled by parameter 1 (in range 0 to 10 seconds) and hold 1 second.

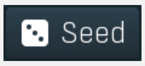
ConvoNoise *(shape;length;hpshape;hpq;hpf;lpshape;lpq;lpf)*
Convolution with white noise is a classic approach for generating impulse response for convolution reverbs. This one provides a noise generator with dampening. It produces in a way an ideal reverberation, without resonances and with optimal decay. However please note that it is technically not algorithmic, so it makes MTurboReverb a sort of hybrid between convolution and algorithmic reverbs. Also note that it may require significant CPU to process and its parameters cannot be automated, since it requires considerable amount of time to update the processing structures.
Shape controls the shape of the decay (in -1 to +1, but bigger numbers are permitted too). Shape 0 provides normal exponential decay, higher values makes the decay stay longer on higher values making it sound longer, lower values make the decay go down faster making it sound shorter and more percussive.
Length lets you override the reverb length. If you don't specify it or enter a negative number, it will follow the global reverb length parameter.
HPShape parameter controls the shape of the frequency of the low-pass filter in time, which simulates high-dampening. It ranges from -1 to +1, but bigger numbers are permitted too. Shape 0 provides the classic linear decay (meaning in the logarithmic spectral profile), so that the frequency of the filter naturally goes down. Higher values makes the frequency stay longer on higher values making it sound brighter. Lower values make the frequency of the filter go down quicker making it sound darker. HPF and HPQ values let you override the filter minimum frequency and Q. By default these are taken from the **Dampening high** panel. Specify -1 if you do not want to override them.
Similarly LPShape controls the shape of the frquency of the high-pass filter, which simulates low-dampening. It's meaning is the same and the LPF and LPQ let you override its maximum frequency and Q.

Example: *convonoise(0;-1;$1;0.2;-1;$2)/example> = convolution reverb with quick decay (shape -1) following the reverb length. The shape of high-dampening filter is controlled by parameter 1 ($1), frequency is not overriden, but Q is set to 0.2. Low-dampening filter shape is controlled by parameter 2 ($2), no other parameters are overriden.*

Transient *(attackdb;release;resolution;link)*
*A fully featured transient processor is useful to smoothen the transients in the reverberation signal. It implements the same algorithm as the level independent processing from MTransient plugin. Attackdb controls the attack shaping and is represented as (x * 20)dB. This is for convenience when using the parameters. Therefore specifying -1 means -20dB gain for attack, 0 means no processing at all, +1 means +20dB attack gain.*
Release lets you manually prolonge the attack section and smoothen the transients even more. It is specified in seconds, default value is 0. Resolution controls the transient detector resolution in seconds, in a way how long the transient needs to be to be detected. You can use it experimentally to adjust the detector to work well with certain signals. The default value of 0.02 (20ms) fits most cases.
Link defines whether the same processing should be applied on all channels. By default it is on (1), so that the plugin keeps stereo

coherence.

*Example: transient(-2) = ultrahard attack attenuation of -40dB (-2 * 20dB).*



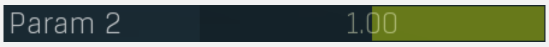
Seed

Seed controls the pseudorandom value generator for the delays and input/output coefficients. When the seed is the same, the generator always produces the same results, hence also the coefficients are the same. By clicking the button you can create a completely new sequence, hence it is an easy way to randomly change the results without messing with complicated parameters the generator is based on. There are billions of combinations the generator can produce.
Range: 0 to 2147483647, default 12345678

Param 1 1.00

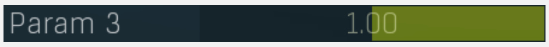
Param 1

Param 1 controls the user defined parameter, which you can implant into the ***Algorithm*** *via '$1'. Note that processing changes to this parameter requires an additional CPU power.*
Range: -1.00 to 1.00, default 1.00


Param 2 1.00

Param 2

Param 2 controls the user defined parameter, which you can implant into the ***Algorithm*** *via '$2'. Note that processing changes to this parameter requires an additional CPU power.*
Range: -1.00 to 1.00, default 1.00


Param 3 1.00

Param 3

Param 3 controls the user defined parameter, which you can implant into the ***Algorithm*** *via '$3'. Note that processing changes to this parameter requires an additional CPU power.*
Range: -1.00 to 1.00, default 1.00

Delay min 5.0%

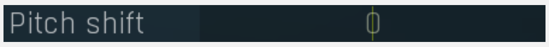
Delay min

Delay min controls the minimum for the delay length used in the generator. This way you can make sparser or denser reflections and control the algorithm in general. If and how are these delay size actually used depends on the ***Algorithm****.*
Range: 0.00% to 100.0%, default 5.0%

Delay max 18.6%

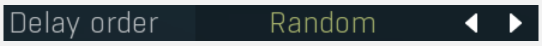
Delay max

Delay max controls the maximum for the delay length used in the generator. This way you can make sparser or denser reflections and control the algorithm in general. If and how are these delay size actually used depends on the ***Algorithm****.*
Range: 0.00% to 100.0%, default 50.0%


Pitch shift 0

Pitch shift

Pitch shift controls the amount of pitch shifting in semitones used in ***Pitch*** *modules. If pitch modules are not used, the parameter will have no effect.*
Range: -12.00 to +12.00, default 0


Delay order Random

Delay order

Delay order controls the order of the delay lengths. On its own it doesn't make much sense, but along with other order settings it lets you set lowest coefficients for smallest delays for example. It also lets you control which parts of the algorithm get which delay lengths. Its effect depends a lot on the algorithm, but think about it this way: an algorithmic reverb is always based on some form of complex delay system. For starters you can think of several delays in series, one after the other. Now if the first delay is short, while the next delays are longer, the first echo will occur very quickly and then the density will get lower. Conversely if the first delay is long, the first echo will arrive sooner and the density will increase afterwards. If they are not sorted, anything is possible. In practice the situation is always far more complex, but the classic advice is try both options and see what suits you best. In most cases sorting provides tigher results, while no sorting results in more ambient results. If and how are these delay size actually used depends on the ***Algorithm****.*

Delay focus -306.4%

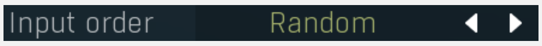
Delay focus

Delay focus controls the tendency of the pseudorandom value generator for the delay system coefficients. Lower values tend to produce shorter delays, which in effect causes quicker increase in the reflection density over time. Conversely, higher values tend to produce longer delays, which then slow down the density. However the output can still become very dense, just in longer time. If and how are the delay sizes actually used depends on the ***Algorithm****.*
Range: -400.0% to 400.0%, default -100.0%

Delay width 100.0%

Delay width

Delay width controls the tendency to produce different delay lengths for different channels, hence to provide wider stereo spectrum as a result of different processing in each channel. It is extended to surround as well. If and how are the delay lengths actually used depends on the ***Algorithm****.*
Range: 0.00% to 100.0%, default 20.0%


Input order Random

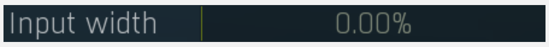
Input order

Input order controls the order of the input coefficients. On its own it doesn't make much sense, but along with other order settings it lets you set lowest coefficients for smallest delays for example. It also lets you control which parts of the algorithm get which coefficients. If and how are these coefficients actually used depends on the ***Algorithm****.*

Input focus 400.0%

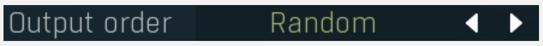
Input focus

Input focus controls the tendency of the pseudorandom value generator for the input coefficients. Lower values tend to produce lower input coefficients, which in effect usually produces less dense output. If and how are these coefficients actually used depends on the ***Algorithm****.*
Range: -400.0% to 400.0%, default 400.0%


Input width 0.00%

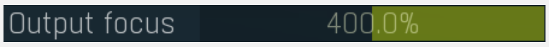
Input width

Input width controls the tendency to produce different coefficients for different channels, hence to provide wider stereo spectrum as a result of different processing in each channel. It is extended to surround as well. If and how are these coefficients actually used depends on the ***Algorithm****.*
Range: 0.00% to 100.0%, default 0.00%


Output order Random

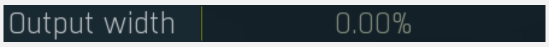
Output order

Output order controls the order of the output coefficients. On its own it doesn't make much sense, but along with other order settings it lets you set lowest coefficients for smallest delays for example. It also lets you control which parts of the algorithm get which coefficients. If and how are these coefficients actually used depends on the ***Algorithm****.*


Output focus 400.0%

Output focus

Output focus controls the tendency of the pseudorandom value generator for the output coefficients. Lower values tend to produce lower output coefficients, which in effect usually produces less dense output. If and how are these coefficients actually used depends on the ***Algorithm****.*
Range: -400.0% to 400.0%, default 400.0%


Output width 0.00%

Output width

Output width controls the tendency to produce different coefficients for different channels, hence to provide wider stereo spectrum as a result of different processing in each channel. It is extended to surround as well. If and how are these coefficients actually used depends on the ***Algorithm****.*
Range: 0.00% to 100.0%, default 0.00%

Panic

Panic

Panic parameter has no direct meaning, but whenever it is changed, the LR module is reset. This may get handy when changing multiple controls via a multiparameter, where each change may cause various audio artifacts due to the delay lines being changed. You can then map this parameter to the same multiparameter, with completely random values, so that it is changed as well and as a result the whole engine is reset and no audio artifacts will occur. The drawback is that it then takes time before the reverb starts sounding the way it should since the delays are empty.
Range: 0 to 2147483647, default 12345678

Channels unrelated

Channels unrelated

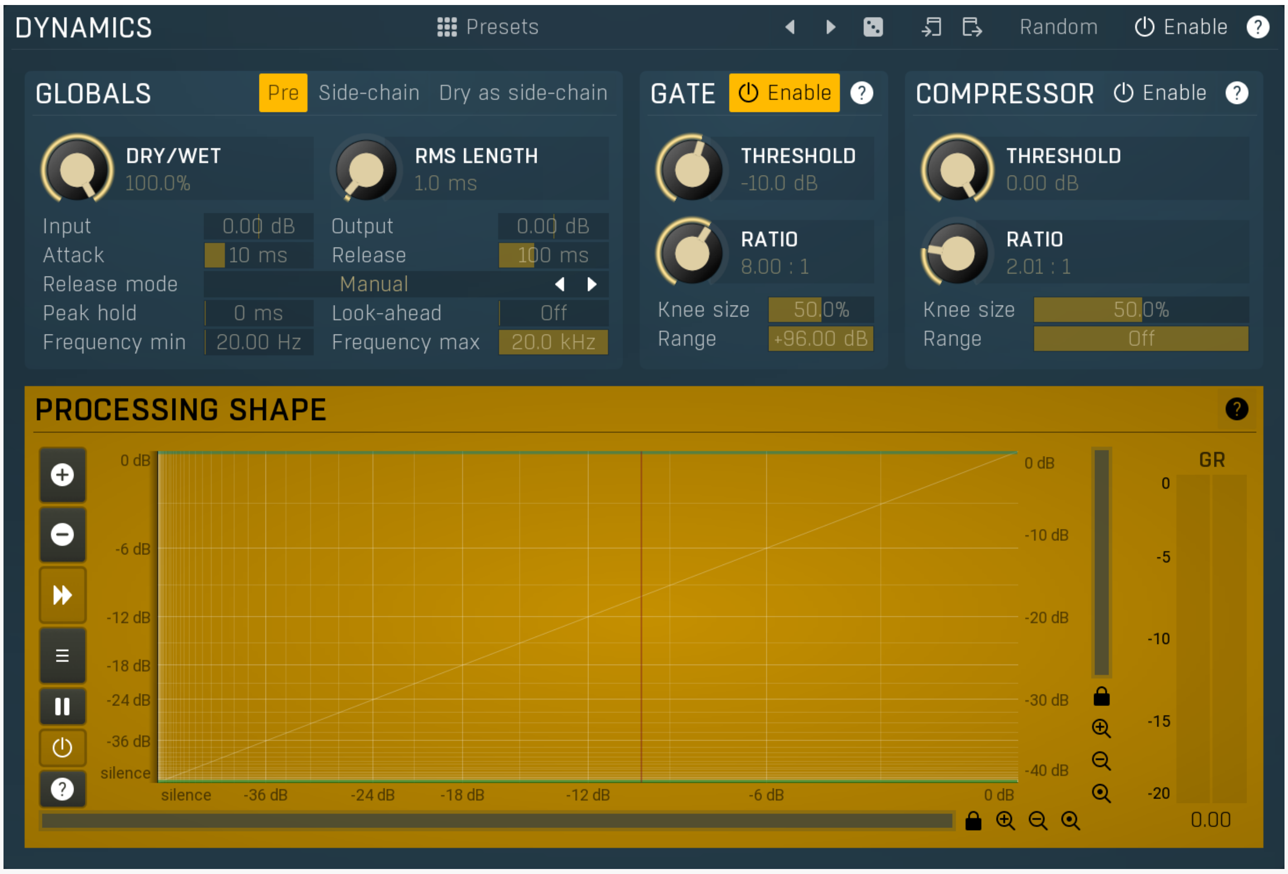
Channels unrelated selects a different algorithm for generating delays for different channels. It will provide different stereo width results, depending on the algorithm and seed.

Channel seeds unrelated

Channel seeds unrelated

Channel seeds unrelated starts randomization for each channel with a different seed. This helps providing a wider image, since different channels will be processed differently. Note that this option does not affect generating delays and input & output coefficients.

Dynamics panel


DYNAMICS
Presets
Random
Enable

GLOBALS
Pre
Side-chain
Dry as side-chain

DRY/WET 100.0%
RMS LENGTH 1.0 ms
Input 0.00 dB
Output 0.00 dB
Attack 10 ms
Release 100 ms
Release mode Manual
Peak hold 0 ms
Look-ahead Off
Frequency min 20.00 Hz
Frequency max 20.0 kHz

GATE
Enable
THRESHOLD -10.0 dB
RATIO 8.00 : 1
Knee size 50.0%
Range +96.00 dB

COMPRESSOR
Enable
THRESHOLD 0.00 dB
RATIO 2.01 : 1
Knee size 50.0%
Range Off

PROCESSING SHAPE

Dynamics panel controls the dynamics processing applied to the input or the reverberation signal (selected using the **Pre** button in the Dynamic Globals panel). This includes gating and compression, both of which are very useful on both signals.

Gating applied to the input signal is a classic technique that lets you reverberate only loud parts of the signal. It is typically used on drums to produce a reverb tail for snare drums for example.

Compressing the input signal lowers the dynamic range creating a "flatter" sound, which creates more stable reverberation without actually flattening the input signal itself.

Gating the reverberation signal is a rather specific effect which lets you abruptly terminate the reverberation signal when its level goes below the threshold. When a long reverb is used, it can often muddy the mix during the parts where the particular instrument isn't playing anymore, or create long endings when used on master, which then need to be faded-out by some other method. Using a gate with threshold low enough can provide an automatic solution which stops the reverb tail sooner than it normally would, removing the muddiness / tail.

Compressing the reverberation signal flattens the reverberation signal and when overused can introduce the typical compression character many are seeking on the tracks themselves, but on the reverberation signal instead. It is mainly useful as a creative tool.

Presets

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

Left arrow
Left arrow button loads the previous preset.

Right arrow
Right arrow button loads the next preset.

Randomize
Randomize button loads a random preset.

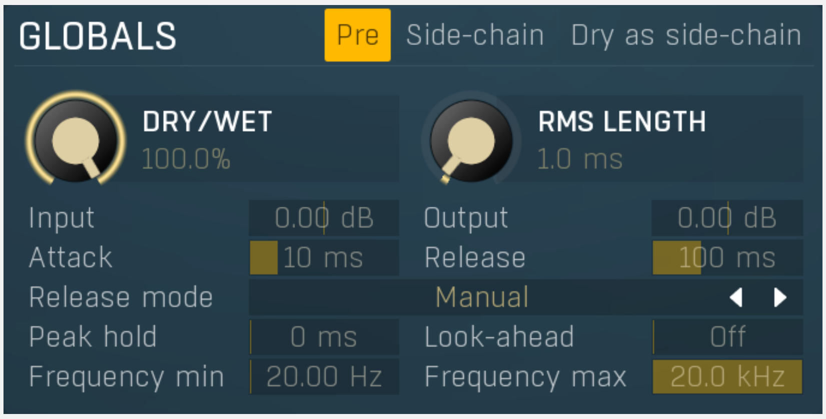
Copy
Copy button copies the settings onto the system clipboard.

Paste
Paste button loads the settings from the system clipboard.

Random

Random button generates random settings using the existing presets.

Globals


GLOBALS — Pre | Side-chain | Dry as side-chain

DRY/WET 100.0%

RMS LENGTH 1.0 ms

Input 0.00 dB

Output 0.00 dB

Attack 10 ms

Release 100 ms

Release mode Manual

Peak hold 0 ms

Look-ahead Off

Frequency min 20.00 Hz

Frequency max 20.0 kHz

Pre

Pre switches between processing the input or reverbation signal. By default it is on, so that the dynamics section is processing the input signal allowing you to produce reveberation only to loud snare hits for example.

Side-chain

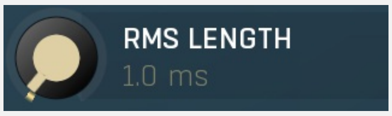
Side-chain button activates the side-chain input as a source for dynamics.

Dry as side-chain

Dry as side-chain button makes the plugin send the dry signal to the side-chain. When **Side-chain** switch is enabled, the dynamics module is listening to whatever you send to the side-chain. When it is disabled however, the dynamics processes the same signal it is listening to. So either it is directly processing the dry input or the reverberation signal. Enabling this option makes the plugin always listen to the dry signal. While that makes no difference when **Pre** is enabled, when it is disabled, the dynamics will be processing the reverbation signal, but listening to the dry input. This can be useful to duck the reverberation while the input audio is loud enough, hence avoiding the 2 signals to collide and let the reverb fill the gaps in the actual performance.

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%



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 1000 ms, default 1.00 ms

Input 0.00 dB

Input gain
Input gain defines gain applied to the incoming signal.
Range: -24.00 dB to +24.00 dB, default 0.00 dB

Output 0.00 dB

Input gain
Input gain defines gain applied to the output signal.
Range: -24.00 dB to +24.00 dB, default 0.00 dB

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 10000 ms, default 10.0 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 10000 ms, default 100 ms

Release mode Manual

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.*

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

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

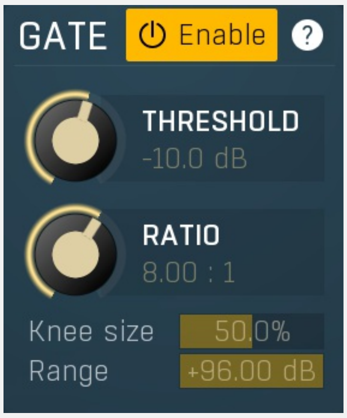
Frequency min

Frequency min defines the cut-off frequency for the detector's high pass filter - minimal frequency.
Range: 20.00 Hz to 20.0 kHz, default 20.00 Hz

Frequency max

Frequency max defines the cut-off frequency for the detector's low pass filter - maximal frequency.
Range: 20.00 Hz to 20.0 kHz, default 20.0 kHz

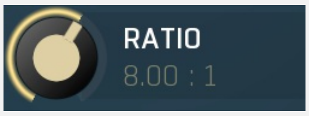
Gate panel



Gate panel contains the controls of the gate processor.

Threshold

Threshold determines the maximum signal level below which the effect starts to apply.
Range: -80.0 dB to 0.00 dB, default -15.9 dB

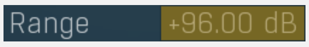


Ratio

Ratio defines the gating ratio of the input signal below the threshold. The higher the ratio, the more brutal the gate will be.
Range: 1.00 : 1 to Infinity, default 8.00 : 1

Knee size

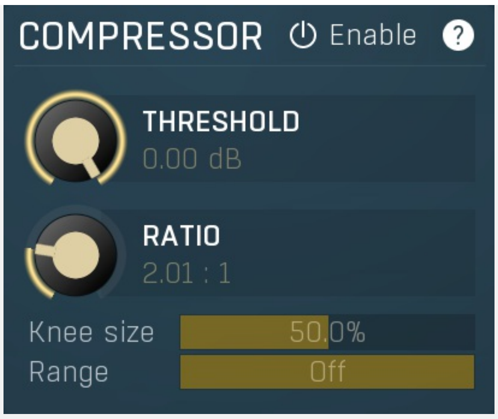
Knee size defines size of the knee.
Range: 0.00% to 100.0%, default 50.0%



Range

Range defines size of the interval above the threshold after which the original signal ratio is restored.
Range: +1.00 dB to +96.00 dB, default +96.00 dB

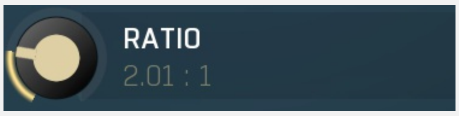
Processor 2 panel



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

Threshold

Threshold determines the minimum signal level above which the compression effect starts to apply.
Range: -80.0 dB to 0.00 dB, default -12.0 dB



Ratio

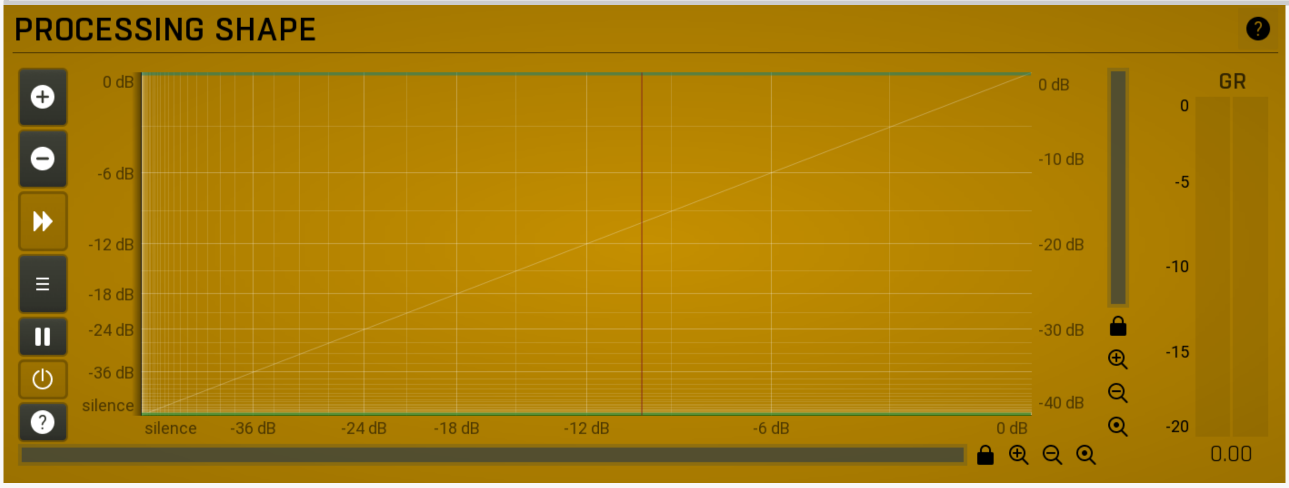
Ratio defines the compression ratio of the input signal above the threshold. The higher the ratio, the more compression you get.
Range: 1.00 : 1 to Infinity, default 2.00 : 1

Knee size

Knee size defines size of the knee.
Range: 0.00% to 100.0%, default 50.0%

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

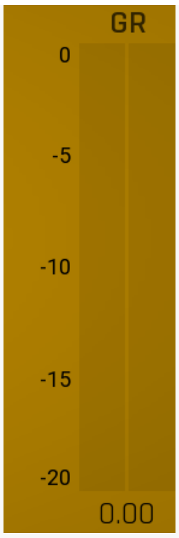


Level shape graph

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.



Meters
Meters display gain-reduction for each channel being processed. Also it contains controls to manipulate time-graphs shown in the transformation shape graph above.

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

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

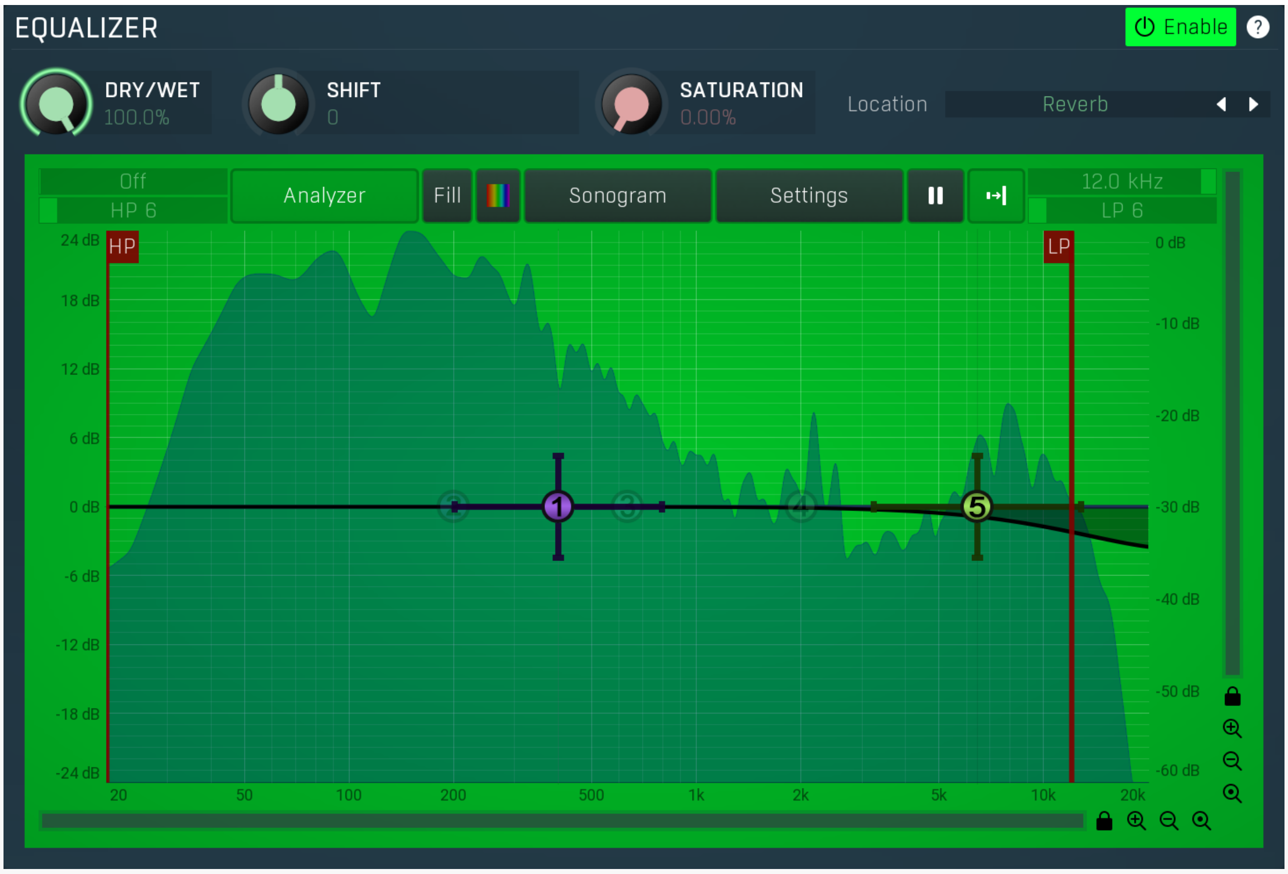
Rewind
Rewind button enables or disables the time-graph static mode. In static mode the graphs are fixed and the current position cycles from left to right; otherwise the graphs move from right to left and the current position is fixed (at the right-hand side).

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

Pause
Pause button pauses the processing.

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

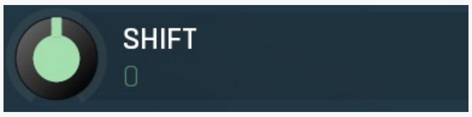
Equalizer panel



Equalizer panel contains the integrated dynamic equalizer you can use to pre-process the input or post-process the output. There are 2 independent equalizers, one processing the standard input channels (depending on the channel configuration, typically left and right, etc.), the other one is processing the mid/side channels, which are transformed from the left and right channels.

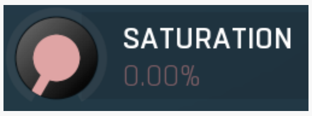
Dry/Wet

Dry/Wet defines ratio between dry and wet signals. 100% means fully processed, 0% means no processing at all. In normal mode only peak and shelf filters are affected correctly, other filters are left at 100% unless the ratio is set to 0%, in which case the equalizer is bypassed.
Range: 0.00% to 100.0%, default 100.0%



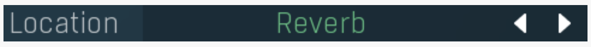
Shift

Shift lets you pitch shift all bands by specified number of semitones. It doesn't change the actual band points, but changes the resulting EQ shape appropriately.
Range: -24.00 to +24.00, default 0



Soft saturation

Soft saturation defines amount of saturation simulating analog equalizers.
Range: 0.00% to 100.0%, default 0.00%



Location

Location controls exactly where in the signal processing chain the EQ will be performed. In most cases the differences will probably be subtle, but bigger differences may occur with more complex settings featuring dynamics processing for example. By default the processing chain is like this:
Input -> Dynamics (if Pre) -> Early & Late Reflections -> Dynamics (if not Pre) -> Widening etc. -> Dry/Wet -> Output

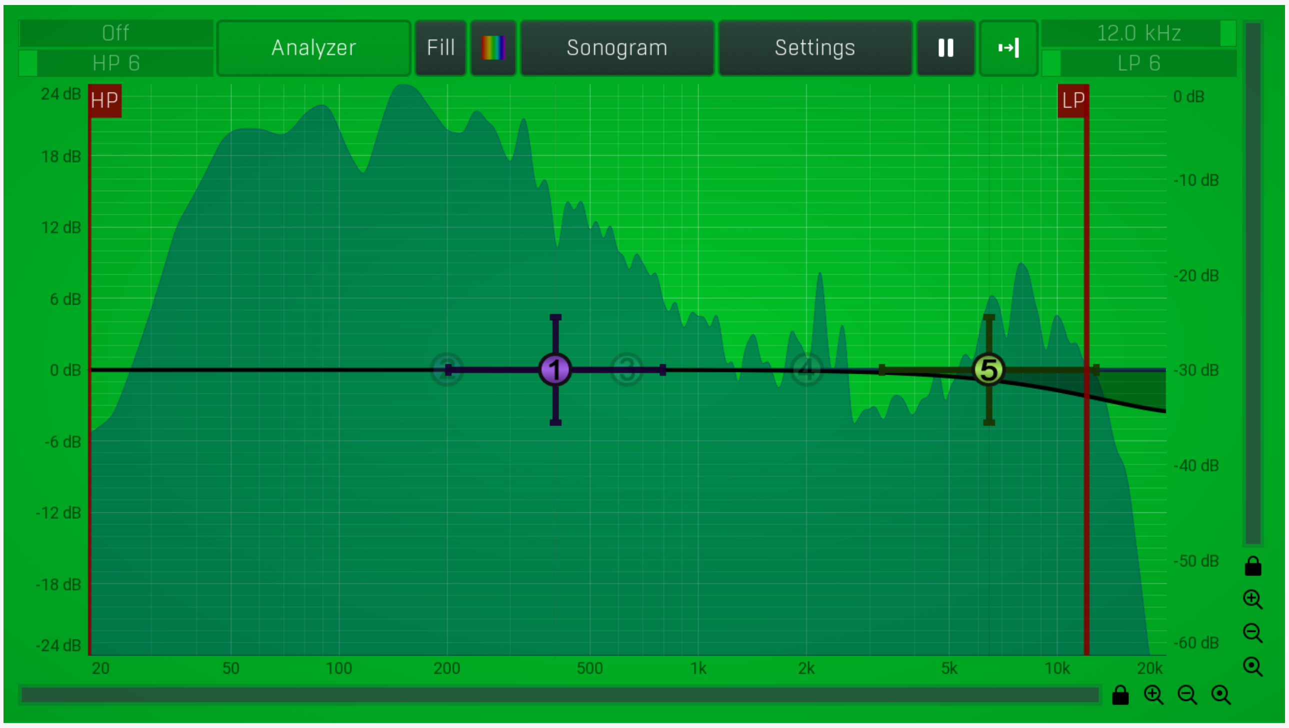
Input mode performs the EQ on the input signal. Hence everything in the reverb is then affected, including dynamics. For example, if you want to trigger the reverb by a bass drum, you would use the **Dynamics** tab as a gate and filter everything above say 100Hz. But if you locate the EQ in front of the dynamics and set it to filter out everything below say 200Hz, which is a common practice, there would be almost no signal left to trigger the reverb.

Input after dynamics mode performs the EQ on the input signal after the dynamics (if located at that stage). Therefore the input dynamic processing would not be affected by the EQ. In the example with the bass drum the reverb input would have filtered frequencies below 200Hz, but the gate would still follow the bass drum.

Reverb mode performs the EQ on the reverberation signal, on the mixed early and late reflections. In most cases the results will actually be the same as the previous mode. The output may be slightly different if you use some kind of nonlinear processing or modulation in the reverb engine.

Reverb after dynamics mode performs the EQ on the reverberation signal after the dynamics (if located at that stage) has processed the reverberation signal, which is the case only if the **Pre** is disabled.

Output mode performs the EQ on the final output signal; in particular, after the global **Dry/Wet**. This means that the dry signal allowed to pass through the reverb would be equalized too.



Equalizer shape graph

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

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

Analyzer

Analyzer

Analyzer button enables or disables the spectrum analyzer, which shows the levels of individual frequencies. In most practical cases it is more convenient to use the sonogram, which shows the frequencies in time, but provides a lower level resolution as the levels are differentiated by color. The spectrum analyzer also provides a micro-sonogram (shown in the bottom of the panel) which uses the same color-based view as the sonogram.

Fill

Fill

Fill button enables or disables the full-sized analyzer micro-sonogram. This means that the micro-sonogram at the bottom of the equalizer graph will fill the whole analyzer view. Color differentiation is often easier to understand than the classical spectrum analyzer, so this might help you better understand the spectrum of your audio material.

An alternative is to use the spectrum sonogram.

Analyzer Rainbow Colors

Analyzer Rainbow Colors lets you see the analyzed sound spectrum in beautiful colors, following the same style as visible light. It ranges from infra-red colors for the lowest frequencies to ultra-violet colors for the highest frequencies in the analyzed audio. If rainbow colors are disabled, the analyzer and graph will be single-colored, following the setup from Settings/Graphs.

Sonogram

Sonogram

Sonogram button enables or disables the spectrum sonogram, which shows levels of individual frequencies in time. Levels are differentiated by color, so the accuracy is not as good as when using the spectrum analyzer. However, the time axis improves the visual orientation in the spectrum for typical audio signals. In contrast, the spectrum analyzer is more of a scientific tool.

Settings

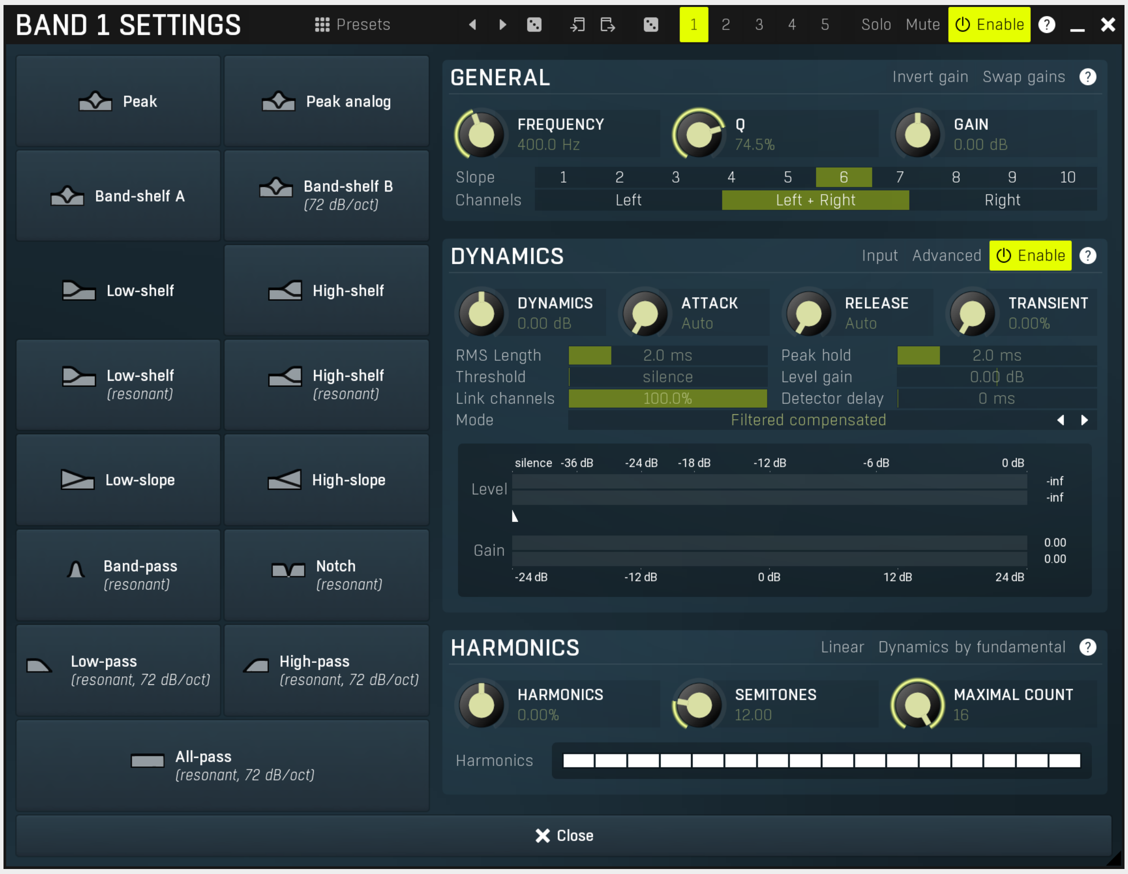
Settings

Settings button shows the settings of the spectrum analyzer and the spectrum sonogram.

Pause
Pause button stops the analyzer temporarily.

Normalize
Normalize button enables or disables the visual normalization, which makes the loudest frequency be displayed at the top of the analyser area (0dB); it does not normalise the sound. This is very useful for comparing frequency levels, however it does hide the actual level.
When comparing 2 spectrums you are usually interested mainly in the frequency level differences. In most cases both audio materials will have different overall levels, which would mean that one of the graphs would be "lower" than the other, making the comparison quite difficult. Normalize fixes this and makes the most prominent frequencies of the spectrum reach the top of the analyzer area (or have the most highlighted color in case of sonogram).

Band settings window



FREQUENCY 400.0 Hz
Q 74.5%
GAIN 0.00 dB
Slope 1 2 3 4 5 [✓ 6] 7 8 9 10
Channels Left [✓ Left + Right] Right

DYNAMICS 0.00 dB
ATTACK Auto
RELEASE Auto
TRANSIENT 0.00%
RMS Length 2.0 ms
Threshold silence
Link channels 100.0%
Mode Filtered compensated
Peak hold 2.0 ms
Level gain 0.00 dB
Detector delay 0 ms

HARMONICS 0.00%
SEMITONES 12.00
MAXIMAL COUNT 16
Harmonics

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

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

Left arrow
Left arrow button loads the previous preset.

Right arrow
Right arrow button loads the next preset.

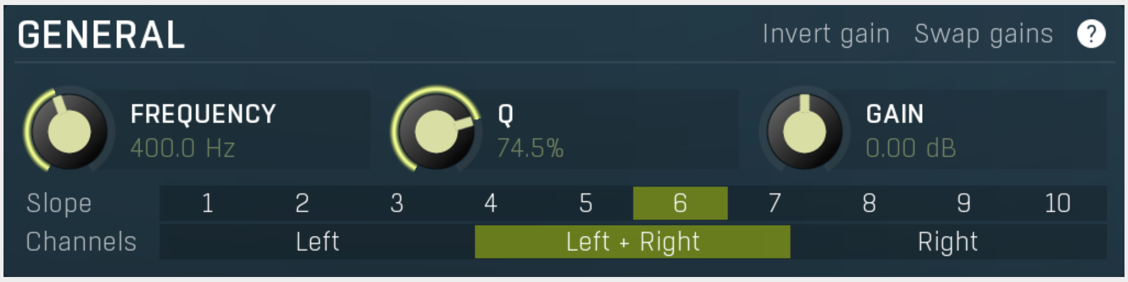
Randomize
Randomize button loads a random preset.

Copy
Copy button copies the settings onto the system clipboard.

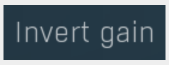
Paste
Paste button loads the settings from the system clipboard.

Random
Random button generates random settings using the existing presets.

General panel



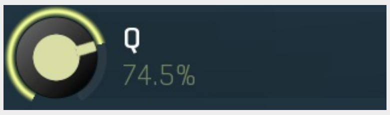
General panel contains standard filter settings such as frequency or Q. Most of these values are available directly from the band graph, but it may be necessary to use these controls for more accurate or textual access.



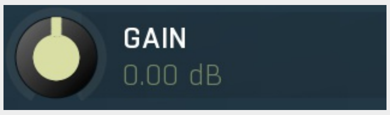
Invert gain
Invert gain inverts the gain of the band, e.g. makes -6dB from +6dB.

Swap gains
Swap gains button swaps values between gain and dynamics gain.

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



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



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

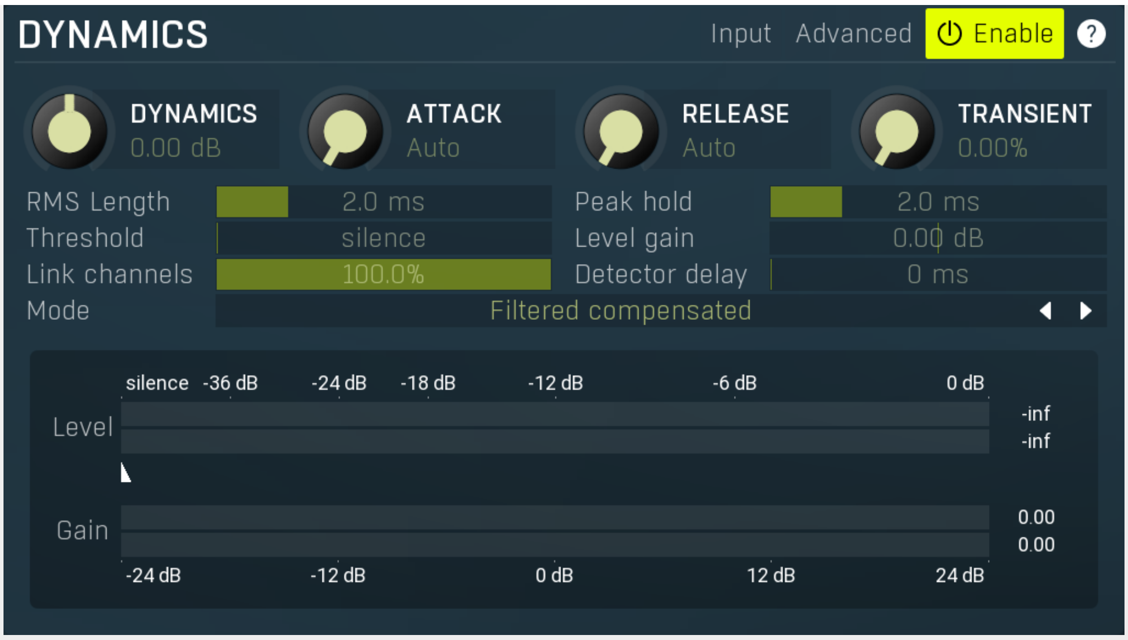
Slope
Slope can potentially duplicate some of the filters creating steeper ones. By default, the slope is 1 and this usually means 2-pole 12 dB/octave filters. By specifying 2 you can make the plugin uses 4-pole 24 dB/octave filters instead etc. To see the actual slope of each filter look into the filter type list on the left.

Channels
Channels controls which channels the band processes. If the input is stereo (left and right channels, L+R, selected on the toolbar **Channel mode** button), then you can make a band process only the left, only the right, or both channels. Similarly when the plugin is set to M/S channel mode, you can choose between mid, side or both channels.

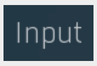
When one of more bands are set to process a single channel, then 2 EQ curves are displayed, in red for the Left or Mid and in green for the Right or Side. If these are not distinct, then we recommend using a style with a light background for these graphs.

You cannot process left with one band and side with the other, because these are working in different encoding modes. In this case you can easily use 2 instances of the plugin in series, one in L/R mode and the other in M/S.

Dynamics panel



Dynamics panel contains settings of the dynamics processing which control how the filter behaves depending on input signal. Normal filters are static, meaning they don't change any features depending on the input signal. If you enable dynamic properties, by making the **dynamic gain** nonzero, the filter will start listening to the level of the input signal. This requires more CPU of course, as such a band is essentially an extremely complex generalized compressor, but the algorithms used are as efficient as it is technically possible.
A dynamic band varies the gain according to the input level. It can listen to the whole spectrum or to just part of it. By default it is driven by the partial spectrum, which it modifies itself, so, for example, when you have a high shelf, it is essentially listening to a high part of the spectrum. You can do many things with such a dynamic processor, but essentially it can work as a compressor or expander. There are many more advanced ideas that you can do and the full power hasn't really been explored yet.

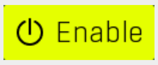


Input

Input switch makes the band measure the input level instead of current level in the chain of bands. When this is disabled (default) and the equalizer is processing the bands serially, which means that each band is processing the output from the previous stage, including level measurement. If you enable this switch however, the dynamic processing will be driven by the original input signal instead.
Please note that when **Side-chain** is on, this switch has no meaning, since side-chain has priority.

Advanced

Advanced button displays additional settings for this band. These contain some more esoteric features, such as a dynamic transformation shape.

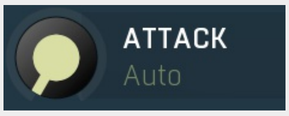


Enable

Enable button enables the dynamic processing. You can use it to switch between enabled and disabled dynamic processing to check the differences.

Dynamics

Dynamics defines the maximum gain of the filter that could be caused by the input signal. For example, if you set it to -24dB and the input signal contained in the band were very strong, the band will be set to an additional -24dB. This would work similarly to a compressor in that band.



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.

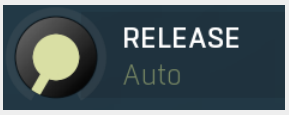
To understand the working of a level detector, it is best to cover the typical cases:

*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.



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.

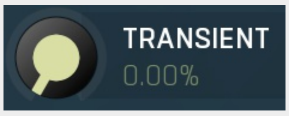
To understand the working of a level detector, it is best to cover the typical cases:

*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.



Transient

Transient lets you mix the level follower output with a transient detector output. This lets you follow signal level, transients or both. Note that since transient level is usually lower than level detector's output, **Level gain** is only applied on the level detector's signal, so you can use this to compensate for the difference in level.

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.

Peak hold 2.0 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.

Threshold silence

Threshold

Threshold controls the minimum level above which the dynamic gain actually starts working.

Level gain 0.00 dB

Level gain

Level gain controls the gain applied to the detector, which can be used for example when the input level is too low, so that dynamic processing would be negligible, unless the level is boosted.

Link channels 100.0%

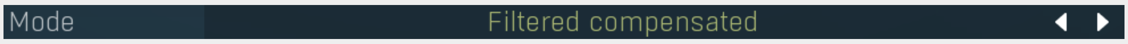
Link channels

Link channels controls how much the signal level for each channel is controlled by the other channels. With 0% the link is disabled and each channel is not affected by the other channels at all. This is suitable to balance stereo channels, for example. With 100% the link is enabled and all channels are controlled by levels of all channels equally (that is the average level of those channels), therefore the processor will apply the same amount of processing on all channels. This is the default in most cases as it preserves relative levels between the channels.

Detector delay 0 ms

Detector delay

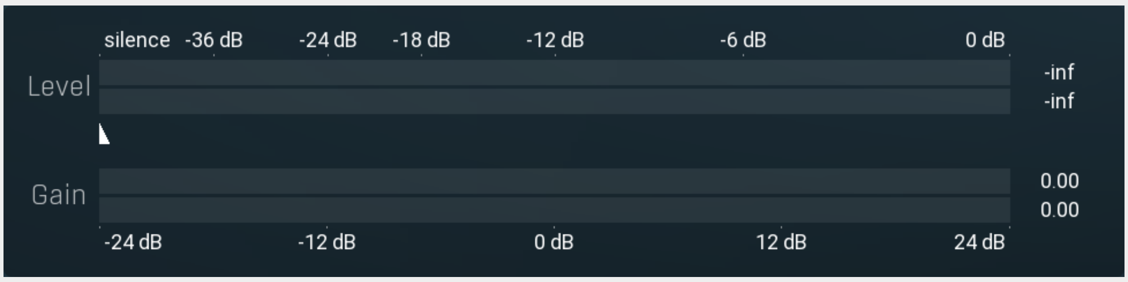
Detector delay lets you delay the detector input, hence the band will react later than the actual input signal.


Mode Filtered compensated

Mode

Mode controls the way the band reacts to the input signal. It has no meaning if the dynamic gain is 0dB.
Filtered compensated mode is default and it means that the source for measuring input level is a filtered signal with additional compensation. For example, when using a low-shelf filter, the signal is low-passed with a filter with the same settings as the low-shelf, therefore the low-shelf filter is affected only by the signal the low-shelf is actually amplifying or attenuating. Since a low-passed signal with cut-off at 100Hz has usually a much lower level than the one filtered with cut-off at 10 kHz, additional compensation is performed to diminish these differences.
Filtered mode is similar, but the compensation is not performed. This may be advantageous for audio materials that do not contain the full spectrum, e.g. a bass line, where the compensation may make things complicated.
Entire spectrum mode is the simplest - it simply takes the input signal without any further processing. This may be useful for example to attenuate selected frequencies when the input level gets too high.

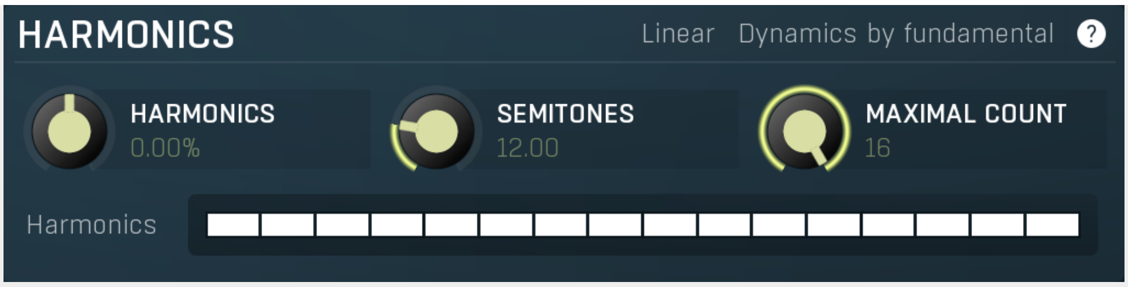
meters



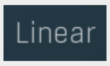
Threshold

Threshold controls minimum level at which the dynamic gain actually starts working.

Harmonics panel

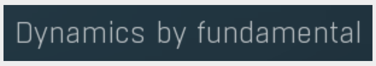


Harmonics panel contains parameters of the harmonics - clones of the main band created at higher frequencies derived from the frequency of the main band. This is often useful for removing natural noises, which usually bring some harmonics with them etc.



Linear

Linear button enables the linear harmonics spacing. When the main band frequency is say 100Hz and the **Semitones** value is 12, then in the default logarithmic mode the harmonics are 200Hz, 400Hz, 800Hz etc., increasing by 12 semitones (1 octave) each time. This is suitable because the filters themselves are logarithmic.
However harmonics generated by physical instruments are not spaced in this way. Rather, for a **Semitones** value of 12, they increase by a multiple of 12/12 of the main frequency each time. For example, for a base frequency of 100Hz, they will be at 200Hz, 300Hz, 400Hz, 500Hz etc. In linear mode the harmonics work in this way, but please note that then there is only a limited set of harmonics and Q is modified to approximate a reasonable behaviour, which is not always possible.

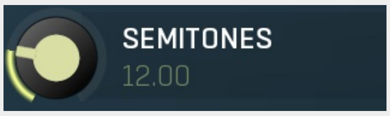


Dynamics by fundamental

Dynamics by fundamental switch causes each harmonic to be driven by the same detector settings as set for the main band. It is disabled by default, which means that each harmonic is literally a clone of the original filter and has its own dynamics detector depending on its own frequency.
Please note that if you want each harmonic to behave in exactly the same way as the main band, you also need to switch on the Input (at the top of the Dynamics panel), otherwise the harmonics would be measuring the signal processed by the main band.

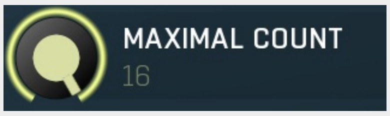
Harmonics

Harmonics defines the gain of the created harmonics. With maximum value (+/- 100%), all harmonics will have the same gain as the main band. A lower value makes the higher harmonics have lower gain. A negative depth will make alternate harmonics have positive and negative gains and is particularly useful for creative effects.



Semitones

Semitones defines the frequency interval of the harmonics. For example, if the band is at 100Hz and the number of semitones is 12 (default), then the first harmonic will be at 200Hz (12 semitones higher), second at 400Hz etc., increasing by 12 semitones (1 octave) each time. Thus they are logarithmically-spaced harmonics. When linearly-spaced harmonics are enabled, this merely changes the ratio between them. In this mode, 100Hz is followed by 200Hz, 300Hz, 400Hz, 500Hz etc, that is, increasing by a multiple of 12/12 of the main frequency each time.
For a value of 7 (a perfect fifth), the logarithmic harmonics would be at 150Hz, 225Hz, 337.5Hz, 506.25Hz etc, increasing by 7 semitones (= 50%, as 1.05946 ^ 7 = 1.498) each time and the linear harmonics would be at 158Hz, 251Hz, 397Hz, 628Hz etc, increasing by 7/12 each time.



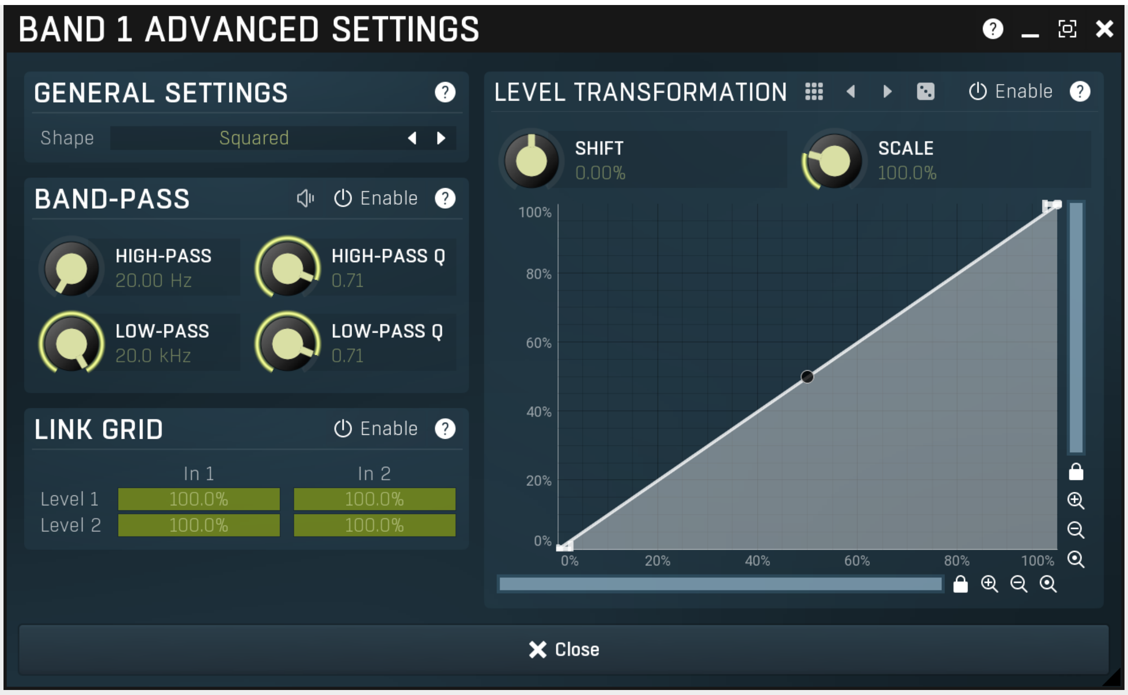
Maximal count

Maximal count defines the maximum number of harmonics that could be created. The harmonics that are created depends on them being activated in the **Harmonics grid**.

Harmonics grid

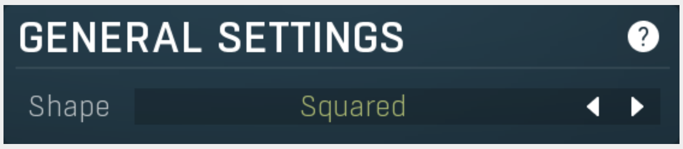
Harmonics grid is useful to turn on/off particular harmonics manually. Click any one to enable / disable it.

Band advanced settings

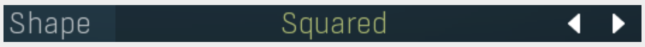


Band advanced settings contains additional settings for the band. These contain some more esoteric features, such as a dynamic transformation shape. It can be displayed by clicking the right mouse button on a band while holding **Ctrl**, from the basic band settings window, or from the band list if provided.

General settings panel



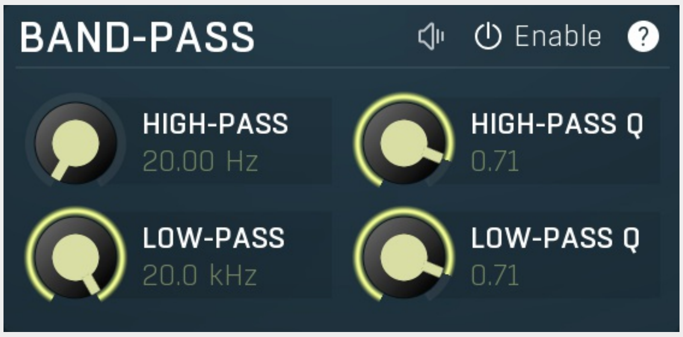
General settings panel contains additional parameters, which are too scientific to be available from the main band settings.



Shape

Shape affects the processing shape. The plug-in features specific non-linear transfer shapes which affect the way the level are interpreted. **Logarithmic** mode is the most physical one, increase from, say, -90dB to -80dB and from -10dB to 0dB produces the same difference in the output dynamic gain. However from the nature of it is tends to generate high gains and usually setting a threshold is needed. **Linear** mode on the other hand tends to stay near minimum gains and usually is the most aggressive. **Squared** mode is a compromise between these two. Comparing the three modes, Linear mode requires the least amount of CPU power and Logarithmic requires the most.

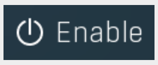
Band-pass panel



Band-pass panel contains parameters of the band pass, which you can use to process the signal that is used measure level of the band additionally. For example, you may want a band at high frequencies to react to bass content; you can do this by placing the band anywhere on the high frequencies and set the low-pass at say 200Hz.

Play

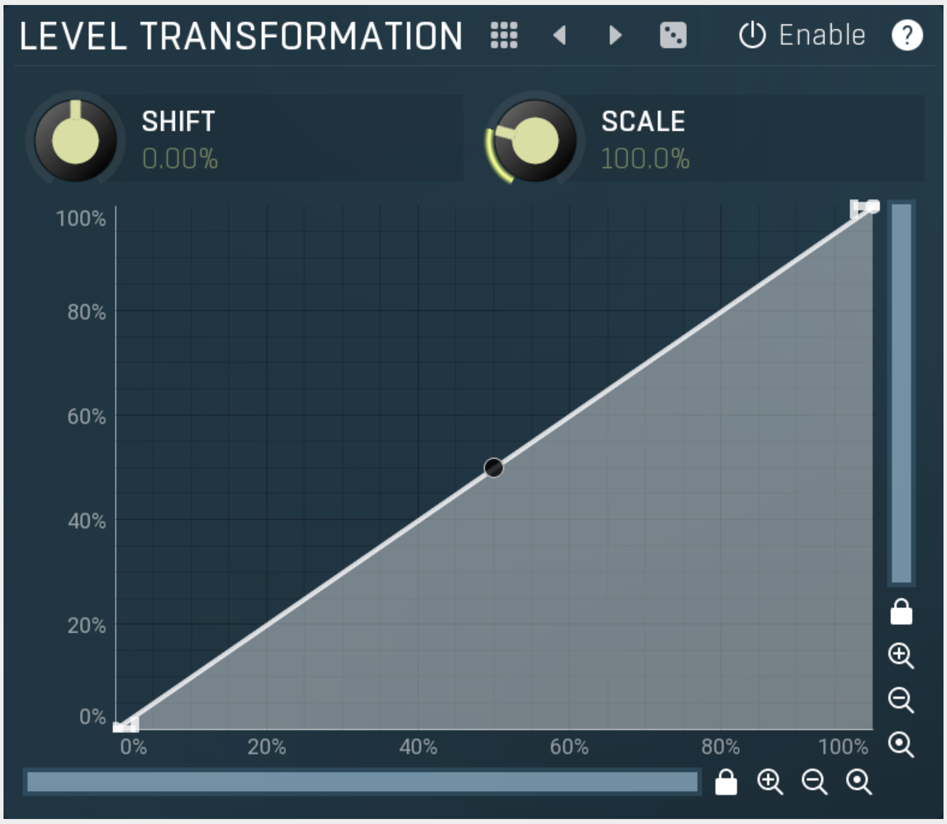
Play button enables the band-pass monitoring and hence could be useful to tweak the band pass.



Enable

Enable button enables the band-pass module. It is off by default to save CPU resources.

Level transformation



Level transformation graph lets you transform the dynamic gain according to the input level. The X axis contains the input level; the Y axis controls the output level, which is then used to set the dynamic gain.

Presets

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

Left arrow

Left arrow button loads the previous preset.

Right arrow

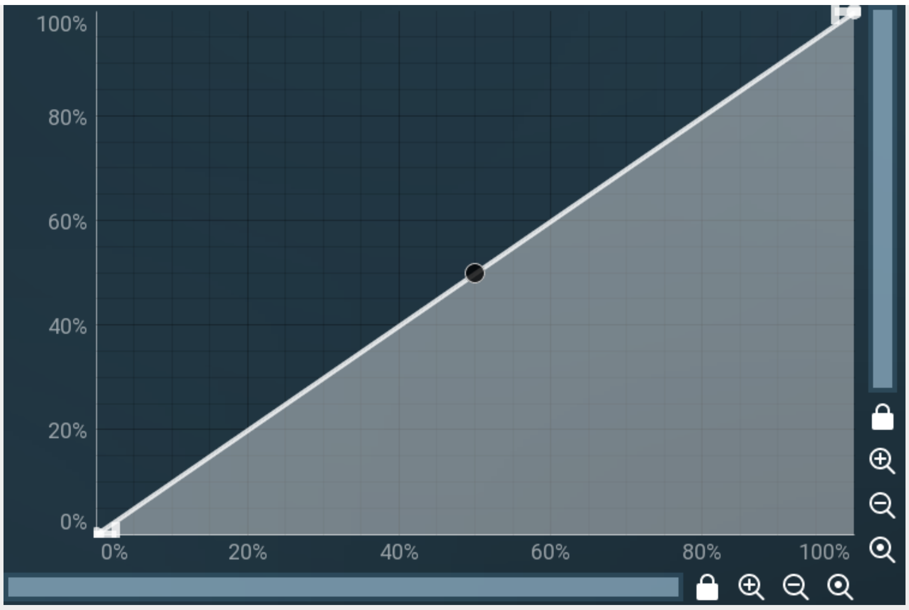
Right arrow button loads the next preset.

Randomize

Randomize button loads a random preset.

Enable

Enable button enables the level transformation module. It is off by default to save CPU resources.

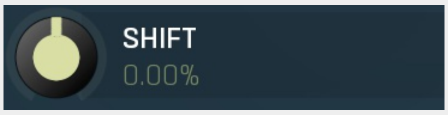


EnvelopeEditorGraph

Envelope graph

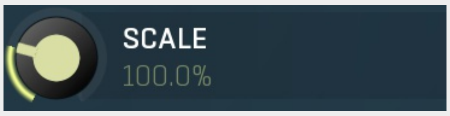
Envelope graph provides an extremely advanced way to edit any kind of shape that you can imagine. An envelope has a potentially unlimited number of points, connected by several types of curves with adjustable curvature (drag the dot in the middle of each arc) and the surroundings of each point can also 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.



Shift

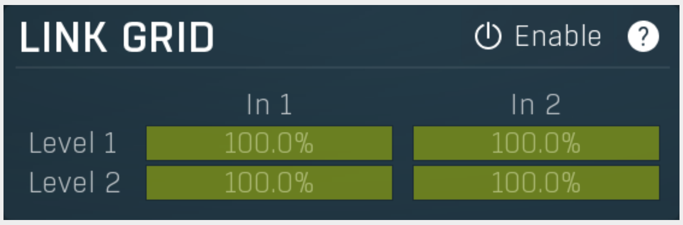
Shift lets you virtually shift the whole graph vertically. This basically shifts the dynamic gain.



Scale

Scale lets you virtually scale the whole graph vertically. This basically scales the dynamic gain.

Link grid panel



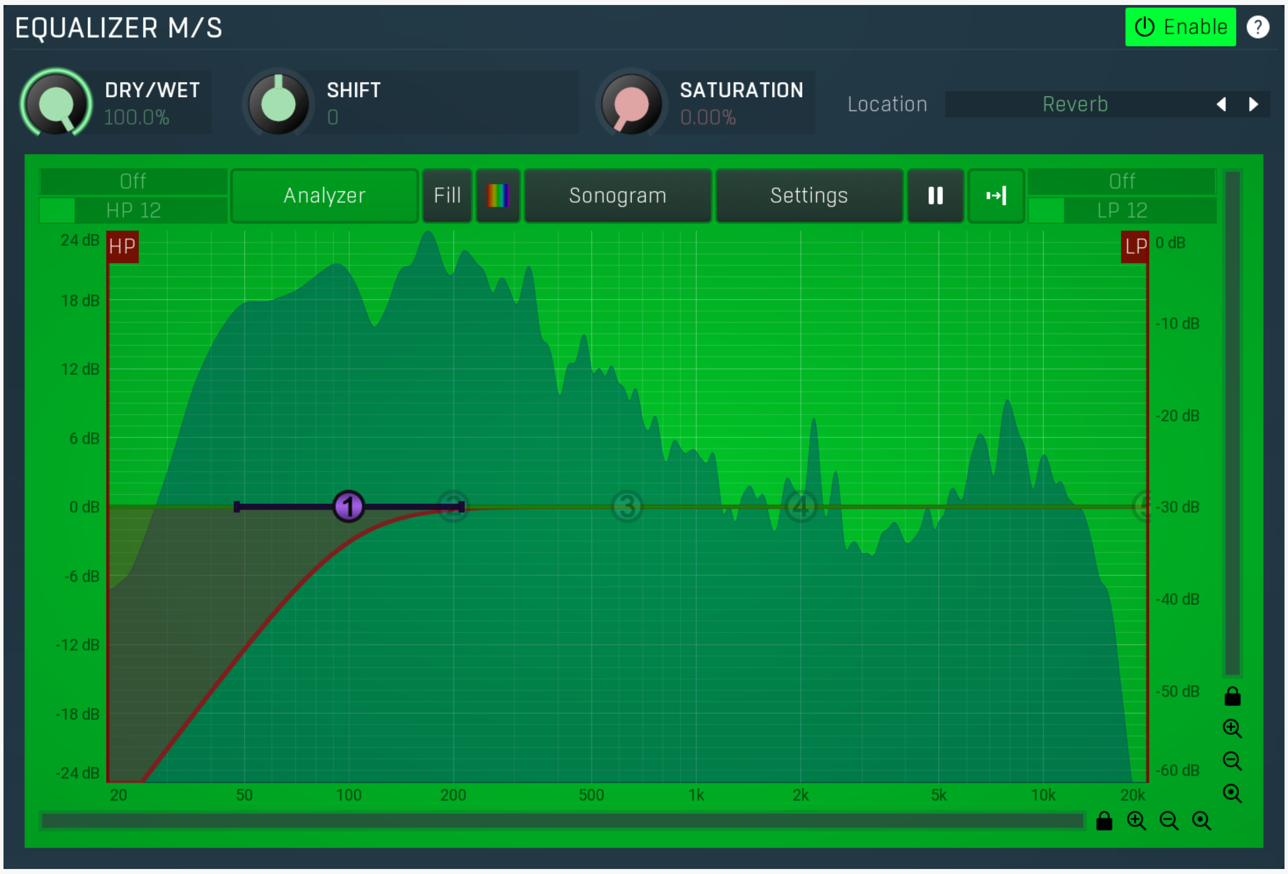
	In 1	In 2
Level 1	100.0%	100.0%
Level 2	100.0%	100.0%

Link grid panel controls the linking between the channels; that is. how the input level in each channel affects the levels in the other channels. By default the way channels affect processing in other channels depends solely on the **Link channels** parameter.
Here you can set up a more complicated relationship. For example, you can make the left channel (1) respond to the right channel (2) only and vice versa. Each column in the grid is an input and each row is an output. Each output level is a mix of the factored input levels. For that example above, the values for "Level 1" would be 0% and 100%, and for "Level 2" they would be 100% and 0%.

Enable

Enable button enables the link-grid module. It is off by default to save CPU resources.

Equalizer panel

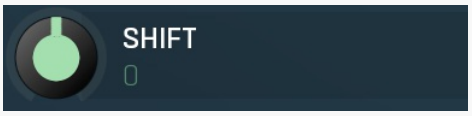


Equalizer panel contains the integrated dynamic equalizer you can use to pre-process the input or post-process the output. There are 2 independent equalizers, one processing the standard input channels (depending on the channel configuration, typically left and right, etc.), the other one is processing the mid/side channels, which are transformed from the left and right channels.

Dry/Wet

Dry/Wet defines ratio between dry and wet signals. 100% means fully processed, 0% means no processing at all. In normal mode only peak and shelf filters are affected correctly, other filters are left at 100% unless the ratio is set to 0%, in which case the equalizer is bypassed.

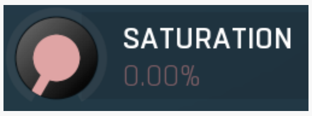
Range: 0.00% to 100.0%, default 100.0%



Shift

Shift lets you pitch shift all bands by specified number of semitones. It doesn't change the actual band points, but changes the resulting EQ shape appropriately.

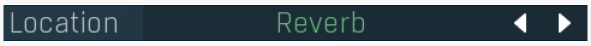
Range: -24.00 to +24.00, default 0



Soft saturation

Soft saturation defines amount of saturation simulating analog equalizers.

Range: 0.00% to 100.0%, default 0.00%



Location

Location controls exactly where in the signal processing chain the EQ will be performed. In most cases the differences will probably be

subtle, but bigger differences may occur with more complex settings featuring dynamics processing for example. By default the processing chain is like this:
Input -> Dynamics (if Pre) -> Early & Late Reflections -> Dynamics (if not Pre) -> Widening etc. -> Dry/Wet -> Output

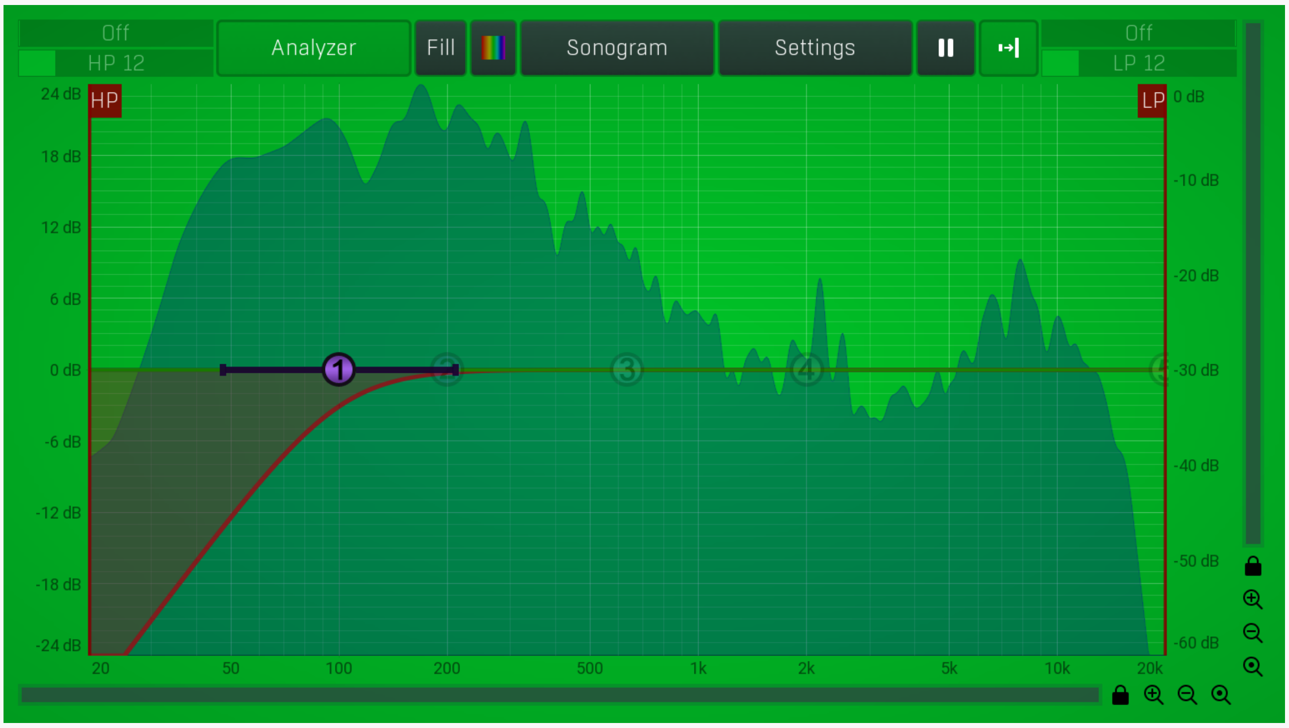
Input mode performs the EQ on the input signal. Hence everything in the reverb is then affected, including dynamics. For example, if you want to trigger the reverb by a bass drum, you would use the **Dynamics** tab as a gate and filter everything above say 100Hz. But if you locate the EQ in front of the dynamics and set it to filter out everything below say 200Hz, which is a common practice, there would be almost no signal left to trigger the reverb.

Input after dynamics mode performs the EQ on the input signal after the dynamics (if located at that stage). Therefore the input dynamic processing would not be affected by the EQ. In the example with the bass drum the reverb input would have filtered frequencies below 200Hz, but the gate would still follow the bass drum.

Reverb mode performs the EQ on the reverberation signal, on the mixed early and late reflections. In most cases the results will actually be the same as the previous mode. The output may be slightly different if you use some kind of nonlinear processing or modulation in the reverb engine.

Reverb after dynamics mode performs the EQ on the reverberation signal after the dynamics (if located at that stage) has processed the reverberation signal, which is the case only if the **Pre** is disabled.

Output mode performs the EQ on the final output signal; in particular, after the global **Dry/Wet**. This means that the dry signal allowed to pass through the reverb would be equalized too.



Equalizer shape graph

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

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

Analyzer

Analyzer

Analyzer button enables or disables the spectrum analyzer, which shows the levels of individual frequencies. In most practical cases it is more convenient to use the sonogram, which shows the frequencies in time, but provides a lower level resolution as the levels are differentiated by color. The spectrum analyzer also provides a micro-sonogram (shown in the bottom of the panel) which uses the same color-based view as the sonogram.

Fill

Fill

Fill button enables or disables the full-sized analyzer micro-sonogram. This means that the micro-sonogram at the bottom of the equalizer graph will fill the whole analyzer view. Color differentiation is often easier to understand than the classical spectrum analyzer, so this might help you better understand the spectrum of your audio material.

An alternative is to use the spectrum sonogram.

Analyzer Rainbow Colors

Analyzer Rainbow Colors lets you see the analyzed sound spectrum in beautiful colors, following the same style as visible light. It ranges from infra-red colors for the lowest frequencies to ultra-violet colors for the highest frequencies in the analyzed audio. If rainbow colors are disabled, the analyzer and graph will be single-colored, following the setup from Settings/Graphs.

Sonogram

Sonogram button enables or disables the spectrum sonogram, which shows levels of individual frequencies in time. Levels are differentiated by color, so the accuracy is not as good as when using the spectrum analyzer. However, the time axis improves the visual orientation in the spectrum for typical audio signals. In contrast, the spectrum analyzer is more of a scientific tool.

Settings

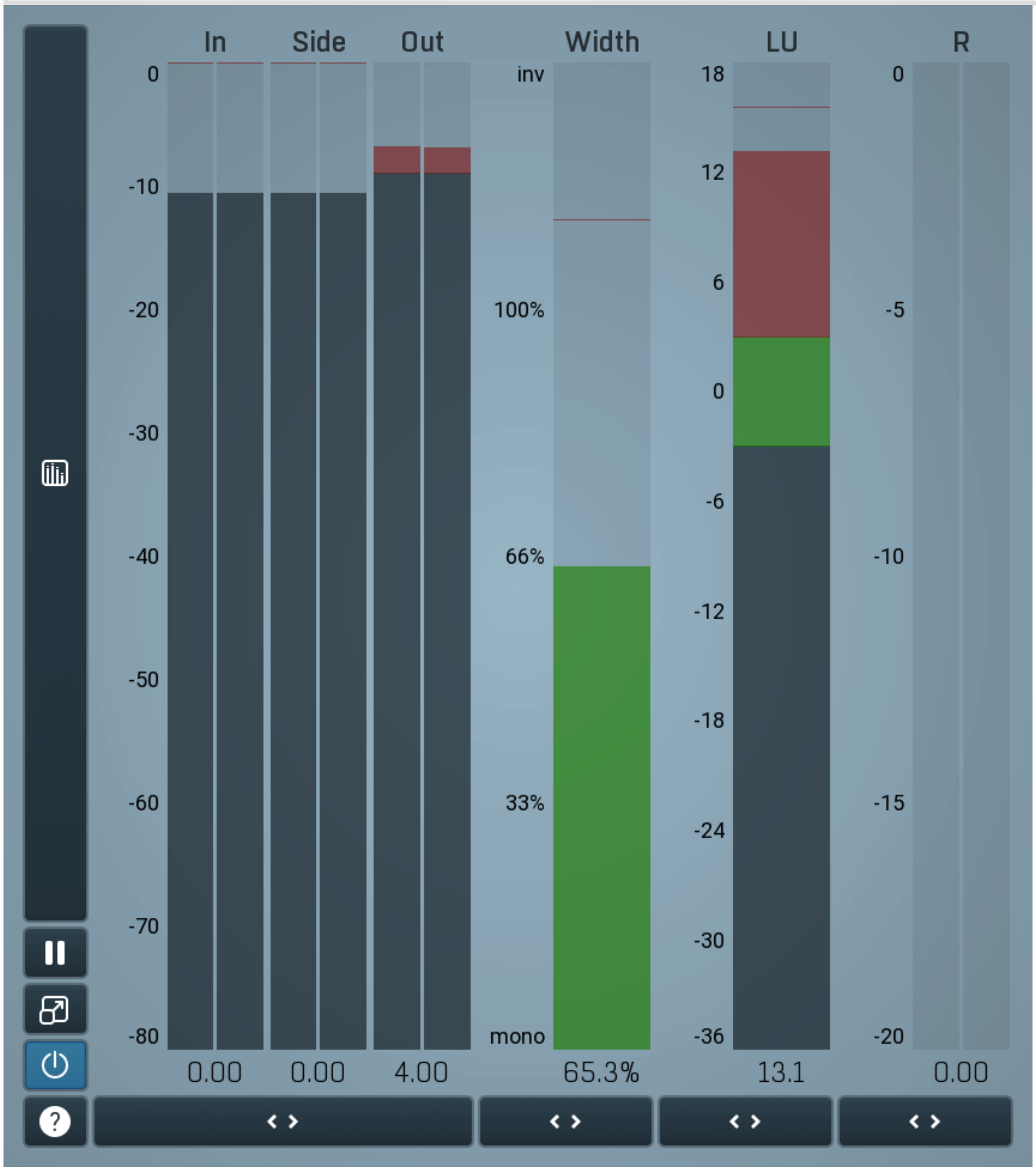
Settings button shows the settings of the spectrum analyzer and the spectrum sonogram.

Pause

Pause button stops the analyzer temporarily.

Normalize

Normalize button enables or disables the visual normalization, which makes the loudest frequency be displayed at the top of the analyser area (0dB); it does not normalise the sound. This is very useful for comparing frequency levels, however it does hide the actual level.
When comparing 2 spectrums you are usually interested mainly in the frequency level differences. In most cases both audio materials will have different overall levels, which would mean that one of the graphs would be "lower" than the other, making the comparison quite difficult. Normalize fixes this and makes the most prominent frequencies of the spectrum reach the top of the analyzer area (or have the most highlighted color in case of sonogram).



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.

LU meter shows the output loudness in EBU-18 scale. The loudness metering follows the ITU-R BS.1770-3 and EBU 3341 specifications. The metering units used are LU (Loudness Units) with 0 LU defined as -23 LUFS (LU Full Scale) and you should consider the LU values to be relative - using them to compare the loudness values between different signals. If the difference in loudness between 2 signals is 10 LU, it is approximately 10 dB as well.

Please note that you should still use your ears to judge loudness properly as there is still no accurate model of human loudness perception and every measurement is only an approximation. Loudness perception is also individual.

If you right click on the meter, additional settings will be displayed. Maximum value displays the maximum since the analysis started, rather than the recent maximum. Loudness pre-filtering uses EBU standard filters to simulate human perception. However, you may want to disable this to get more technical measurements.

There are 3 types of loudness measurements, all following the EBU specifications.
Momentary loudness uses an RMS sliding analysis window of 400 milliseconds; therefore it shows quick fluctuations in loudness.
Short-term loudness works in the same way, but uses a window of 3 seconds, therefore it provides more stable loudness measurements.
Integrated loudness shows the overall loudness, hence it is affected by the whole track from the beginning of the playback until you reset it by clicking on the value field. The host may reset it too; it depends on your host.

Please note that the **Integrated loudness** is NOT the same as an averaged loudness, as it ignores quiet passages. Imagine a track which is generally quiet but has a few loud sections. The averaged loudness will be less than the Integrated loudness. Its calculation uses gating to ignore those quiet passages (levels less than 10 LU less than the current ungated level) of the track. Essentially, **Integrated loudness** is a measure of the loudest sections of the track.

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.

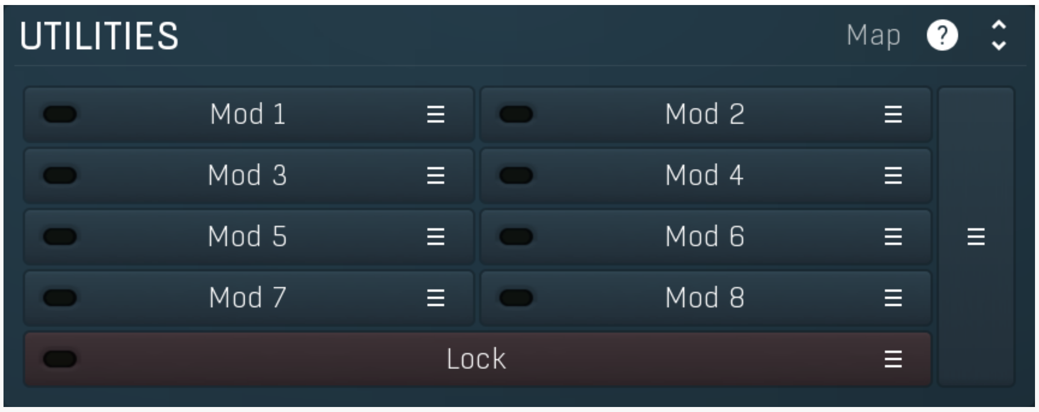
Collapse
Collapse button minimizes or enlarges the panel to release space for other editors.

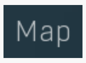
Collapse
Collapse button minimizes or enlarges the panel to release space for other editors.

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

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.