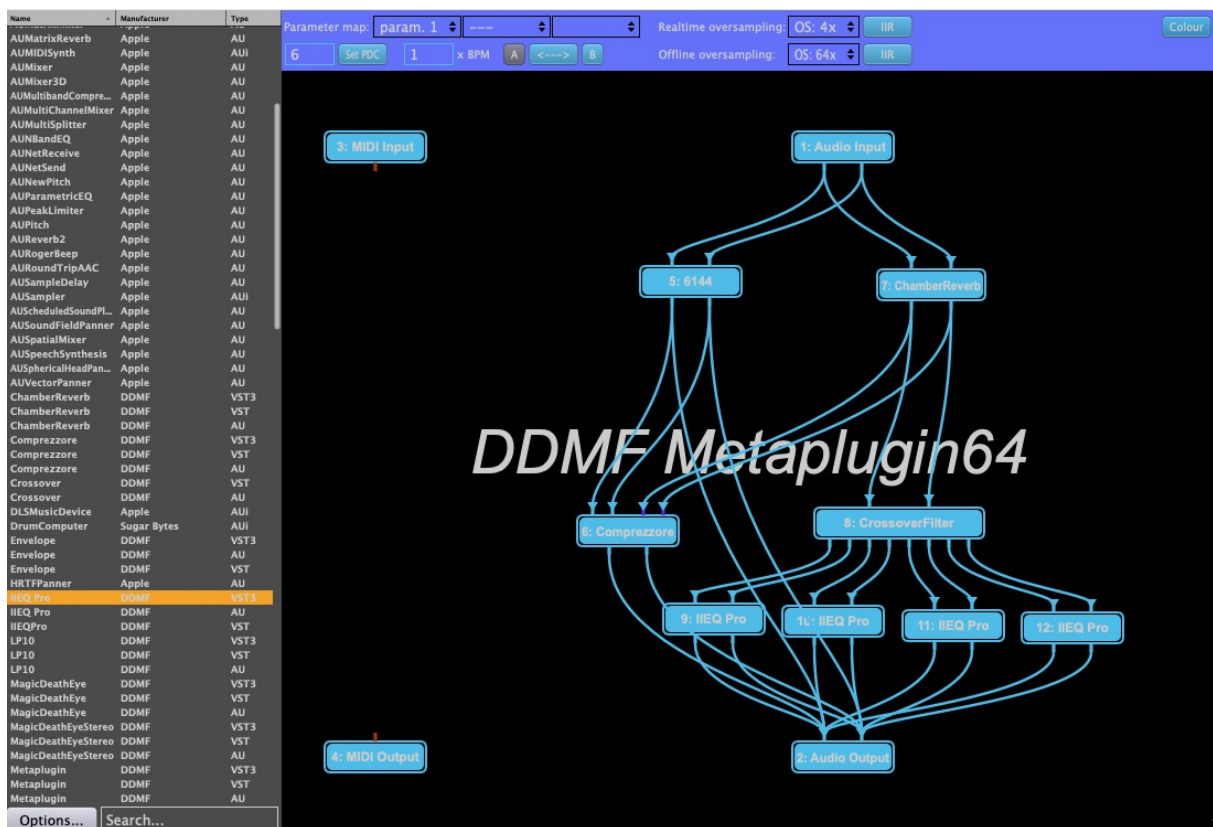


DDMF Metaplugin manual

Version 4.3.6

Thank you for using DDMF's Metaplugin! The following should help you to get the most out of this audio plugin. The possibilities are enormous but the principle is actually very simple. If, after reading, you still have questions about the usage of this software, don't hesitate to contact support@ddmf.eu !



Setup:

Metaplugin comes as a (Windows or Mac) installer package. The available plugin formats are VST, VST3, AU (Mac only) and AAX. You can choose which of these formats you want to have installed. On Windows you can also choose the VST folder where you want the 32 and 64 bit VST versions to be copied to. After installation your host should recognize Metaplugin, as well as MetapluginSynth, the instrument version of Metaplugin. If not, you might have to perform a manual rescan of your plugin folder. Please refer to your host's operation manual on how to do this.

Usage:

Metaplugin produces no sound of its own, but can load other VST//VST3/AU plugins (both synths and effects) and chain/connect the in- and outputs of these effects. This way complex effect arrangements can be created which are not available as a single effect plugin. All created configurations can be saved for repeated usage.

Let's look at a typical example as shown in the image above: loaded plugins are represented by small rectangles, with the name of the plugin written onto it. The input channels of the plugin are located above the rectangle, the output channels below them. Channels of different plugins can be connected by dragging the mouse from an output to an input channel. The rectangles named "Audio Input" and "Audio Output" are always present and represent the in- and output channels of Metaplugin itself. The "Midi Input" contains the midi signals that are routed to Metaplugin in your host. Midi pins and cables are drawn in orange, to distinguish them from the audio part.

Plugin organizer:

On the left hand side of the UI, you find a plugin organizer. Upon first load, it will be empty, and you will be asked to scan for plugins via the "Options" button. Use it to scan for VST, VST3 or AU (OSX only) plugins. When operating in a 64 bit host, you'll also have the option to scan for "bridged" VST or AU plugins. Use this option to scan for plugins that are only available in 32 bit format. Once the plugin organizer is populated with some plugins, you can simply drag and drop them onto the main area to the right, and start wiring them together. Use the "Search" field to restrict the displayed plugins to only those instances whose name or manufacturer match the entered search string.

User presets:

Also on the left side, as a second tab next to the plugin organizer, there is a preset manager. The preset manager lets you store and reload user presets independent of the host/DAW you are using. They will be stored in a system-wide location so that you can if you load Metaplugin in a different host on the same machine, the presets will already be there. To store your current plugin graph as a new preset, use the "Save new preset" button. You will be asked to give your new preset a name, and it will show up in the preset list. Right-clicking on an already existing preset lets you rename it, resave it (if you have changed something) or delete it from the list. Also, you can assign it to one of the available categories. For this, you first have to create at least one category (e.g. "My bass presets"). You can add categories to your preset list by clicking the "New category" button. Once categories are present and filled with at least one preset, you can browse them in a folder-like structure. Categories with zero presets in them will no longer be shown when reloading your preset list upon the next loading of Metaplugin.

Autoconnect:

It can be cumbersome to connect all those cables for each and every plugin you want to load, and most of the times, you will only want to connect corresponding in- to output channels,

without any specific routing setup. For this case, there is an autoconnect option: when you drag in a plugin from the organizer, and have the CTRL key pressed when releasing the mouse button to finish the drag, Metaplugin will automatically connect the newly loaded plugin to the nearest plugin above and below it. The connections will be made such that all channels with the same channel index are connected until at least one of the plugins to be connected runs out of channels. So if your new plugin is a stereo-in-stereo-out-plugin, and the plugins above and below have two out/input channels, respectively, ordinary stereo connections will be made. But if the upper plugin has, say, 8 channels, but the new plugin only three, the first three channels of the upper plugin will be connected to the three channels of the lower plugin. Simply try it out and it will be self-explaining. Of course, you can always change the connections after the auto-connect operation.

Alternative load method:

There are two alternative ways to load plugins into Metaplugin: one is by dragging and dropping a plugin file onto Metaplugin's user interface; another is by right-clicking onto Metaplugin and choosing "Load plugin". A file dialog will appear which will let you load one of the plugins available on your computer. Finally, plugins that have already been loaded before in a session appear as possible choices in the menu that appears after right-clicking. Also, audio inputs and audio outputs can be added that way. Of course, in principle you can use just one audio input or audio output element, but when the networks become more complex it is often easier to overlook the structure with more than one input or output element.

Plugin bridge:

Metaplugin includes a built-in bit bridging solution. When running in a 64 bit host, Metaplugin employs an internal 32-64-bit bridge in order to be able to load 32 bit plugins. This way you can use older plugins for which a 64 bit version is maybe not available, while still getting all the advantages of a 64 bit host. Bridging in the inverse direction is possible, too: if you have an older 32-bit host but want to use 64 bit plugins in it, you can use Metaplugin for this purpose. Please note that, on Mac, with the introduction of Catalina any support for 32 bit processes has been discontinued. So the bridge option on Mac is only available up until OSX Mojave.

There is a second bridging option for Mac users on newer computers with Apple Silicon chips: Metaplugin can run as a native plugin in an Apple Silicon host, and load plugins in Intel format for which a native version has not yet been released (or maybe never will). Along with the BridgedVST/BridgedVST3/BridgedAU formats from the previous paragraph, you will find BridgedIntelVST/BridgedIntelVST3/BridgedIntelAU for this purpose when selecting the plugin format to scan for. As with any other plugin, you can also drag and drop those plugins directly onto Metaplugin's UI.

General commands

When double-clicking onto one of the rectangles representing a plugin, the factory graphical user interface of that plugin will appear in a separate window. Here you can change the plugin's settings in the usual way. Of course, Metaplugin remembers all these settings when a project is being saved in your host (not true for the demo version ☺).

Another option that appears on right-clicking is "Load graph" and "Save graph". Let's say you have created the perfect effect chain for the singer of your band, consisting of an EQ, two compressors and a pitch shifter. Instead of having to load all the plugins in this chain again for every new song, you can simply save the chain/graph in a .metaplugin file and load it whenever you need it.

Every loaded plugin gets an ID, where 1, 2 and 3 are given to Audio in/out and Midi input, so the first loaded plugin will usually get ID 4. This helps to distinguish between multiple instances of the same plugin. The ID is displayed in the small green rectangles and in the header of the GUI of each plugin.

The interface is resizable via the little triangle in the lower right corner. Just click on it and move the mouse around to make as much space for your networks as needed.

Configure plugin audio I/O

Another option when right-clicking on a plugin rectangle is to configure the I/O channels of a plugin. Most plugins offer various options to setup their input channels, and sometimes even have the possibility to add sidechain channels. By default, Metaplugin shows the maximum number of the available channels (so, for instance, if a plugin can deal with mono and stereo signals, Metaplugin will show two input and output pins). With the audio I/O menu you can change the number of channels in the audio buses of a plugin, or add/remove an optional bus entirely. This can be helpful to save CPU power when you only plan to use a plugin in mono, or you decide you do not need a plugin's sidechain channels.

Variable send amount

In Metaplugin you can change the Wet/Dry ratio of the signal going through each of the loaded plugins. You can do this by Shift-dragging the mouse horizontally over a plugin rectangle. A full rectangle means 100 % wet signal, while the slider pulled all the way to the left means the plugin will be bypassed completely.

Channels

Metaplugin supports up to eight parallel channels, plus two sidechain inputs. The number of actually available channels is a bit different for the VST version compared to the Audio Unit version. The VST version will always show all eight channels at input and output, even when Metaplugin is loaded into an ordinary stereo channel (the first two pins then contain the stereo signal, the remaining four are empty). When using the Audio Unit version, Metaplugin will only show two pins at input and output when loaded into a stereo channel, but six when, e.g., loaded into a 5.1 channel.

PDC (Plugin Delay Compensation)

Some plugins, due to the internal structure of their audio algorithm, cause a delay of a certain number of samples (latency). Most hosts compensate for this delay so that tracks with different latency are played back in sync. Metaplugin takes care of Plugin Delay Compensation as well. This means that you can build a graph of plugins with different latencies without having to worry about timing issues. The total latency is reported to the host dynamically.

There's one caveat, however: when using more than one "Audio Output" unit and creating graphs with different latencies for each of these output units, the PDC algorithm will not work. In other words, proper delay compensation is only guaranteed when using a single output unit.

The overall PDC is reported in the upper right corner (in samples). Here, you can also set the overall PDC that Metaplugin reports back to the host by entering an arbitrary number and pressing "Set PDC".

Adjusting the BPM ratio

Metaplugin reads the beats per minutes (BPM) from the host and transmits it to the loaded plugins. Temp-synced plugins will play at the right playback speed. But you can also adjust the BPM ratio freely: let's say your project BPM is 120 but you want your drum loop to play with only 60 BPM. All you have to do is enter "0.5" in the "x BPM" entry box. Now every plugin that is being loaded will play back at half the host's BPM. Please note that you have to load a plugin anew after adjusting the BPM.

Oversampling

Metaplugin can operate the whole plugin chain in realtime mode with up to 16x oversampling, and in offline mode (during rendering) even with up to 64x oversampling. The process of oversampling consists of increasing the sample rate in order to avoid possible artefacts when, e.g., harmonics are being generated that are higher than the maximum frequency that is supported by the current sample rate and will therefore fold back into the usable spectrum, creating unpleasant results (look up "aliasing" if you want to know more about this).

Select your desired oversampling factor from the dropdown menu (OS: off is the default, no-oversampling state). You can select the filter that is being used by the oversampling algorithm, using the “IIR/FIR” button right next to both the realtime and the offline oversampling dropdown menu. “IIR” uses a minimum phase filter, with very small latency but a noticeable phase change at Nyquist frequency. “FIR” causes a slightly higher delay, but comes with linear phase response.

CAUTION: 8x oversampling or higher is quite a mouthful for most plugins, especially if you are already operating at a base sample frequency higher than 44.1 kHz. It is absolutely possible to experience audio dropouts, stuttering or even crashes when operating a plugin outside of its designated frequency range. And although you might think that “anything goes” in offline rendering, it could still be the case that a plugin's internal algorithm is not built to handle frequencies in the MHz range. Don't blame it on Metaplugin, and take it down a notch, 4x oversampling is usually sufficient to hear a noticeable improvement in nonlinear plugins.

A/B option

Metaplugin has two internal states, the “A” state being active by default. In the usual fashion, you can work on two different chains to make direct comparisons between differing setups. Click on “A” or “B” to switch to the respective state, and click on the “<-->” button between the “A” and “B” button to copy the state from the currently active to the currently inactive states.

Scan in background

By default, when you scan for new plugins either with the plugin manager or by dragging and dropping a plugin file onto the UI, the scanning is performed in a background process in order to not make Metaplugin crash when there is something wrong with the scanned plugin. But a few plugins, for obscure reasons, fail the scanning process that way, so this is an option to try when a plugin is not being recognized by Metaplugin using background scanning. You also need to scan in background when running Metaplugin in LogicX in newer Macs!

Bypassing a plugin

Right-click on one of the rectangles representing a plugin, and use the Bypass option to activate or deactivate audio processing by that particular plugin.

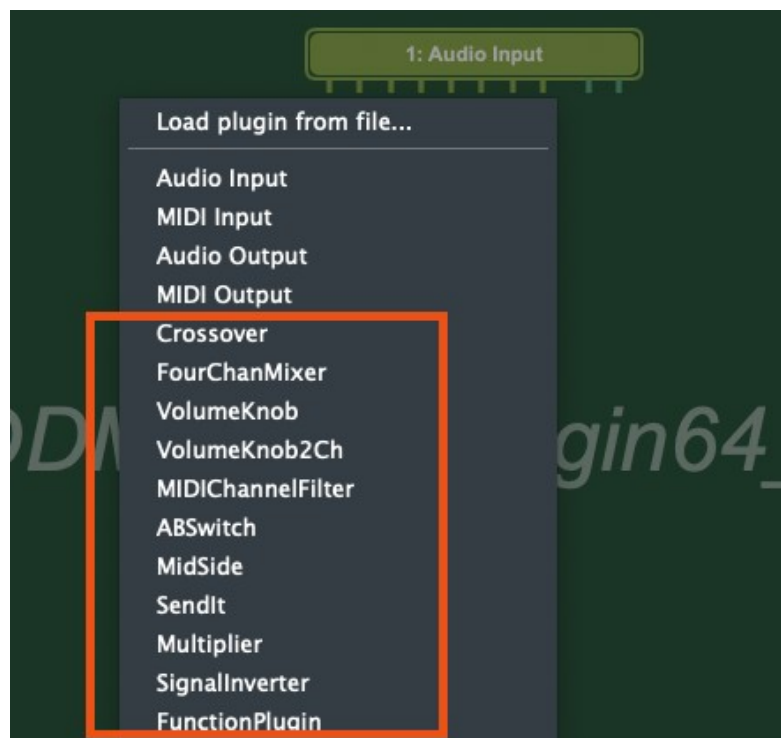
Accessing plugin parameters

Metaplugin exposes 100 parameters to the host. These parameters can be mapped onto the parameters of loaded plugins using the three parameter mapping drop-down menus in the top row. For instance, if you have loaded an effect called SuperEQ which has an internal parameter called HighGain and you want to map that onto parameter1 of Metaplugin, simply set the left drop-down menu to parameter1, the middle one to SuperEQ and the right one to

HighGain. Now, when you automate parameter1 in your host program you will actually automate the HighGain parameter of SuperEQ...

Internal plugins

As an extra bonus, a bunch of "internal" plugins are included in the package that were specifically designed for what we think are typical usages of Metaplugin. Starting from Metaplugin 4, those plugins are no longer installed on your harddisk, but included in Metaplugin itself. You can access them via right-clicking on an empty spot in the graph editor, in the section below the in- and output modules:



Plugins without UI/editor:

VolumeKnob (which comes in a 1 channel and a 2 channel version), SignalInverter, Multiplier, MIDIChannelFilter and ABSwitch are very basic helper plugins with their graphic control already a part of the "plugin rectangle", i.e. they will not show a separate control editor when double-clicked.

VolumeKnob is a simple volume filter to change the volume of a single or a pair of connections.

Multiplier takes two signals as input and outputs the product of them.

SignalInverter is another mono module that allows you to invert a single channel (phase shift of 180 degrees).

ABSwitch has a stereo input bus and two stereo output buses, and allows you to try out two different routings in your chain with a single click.

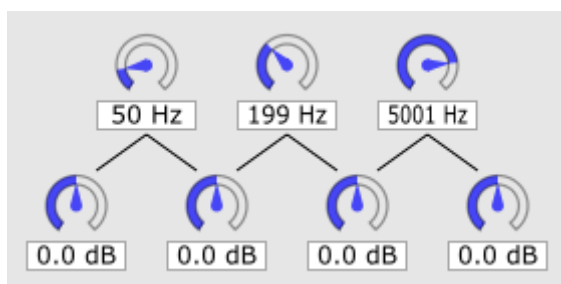
MIDChannelFilter takes a MIDI signal as input and outputs another MIDI signal with events only on the selected channel.

IMPORTANT: the controls of those plugins without editor will only be responding to your mouse inputs when you hold the SHIFT button down. If you are simply trying to perform a drag or click motion without SHIFT, they will simply be dragged around like any other plugin rectangle.

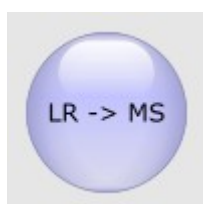
Plugins with UI/editor:

In the Crossover plugin you find a flat-summing four band crossover filter using Linkwitz-Riley filters. The FourChanMixer has 4 stereo input buses and one stereo output bus, with the UI consisting of volume controls for the four input buses. In MidSidePlugin there is a Mid-Side matrix which can encode a stereo left-right signal into a mid-side signal and the other way round. Finally, with SendIt you can send audio signals across tracks in your digital audio workstation in case it doesn't offer sufficient routing possibilities (for side-chaining, for instance).

The GUIs of these helper plugins are stripped down to the absolute minimum, but especially the crossover filter gives you a lot of control with the few knobs it has. In the first row, the frequencies that divide the four bands can be set. The second row contains the gain stages for each of the four frequency bands. The plugin has two “In” and eight “Out” channels (the latter ones being the left and the right channel of each of the four frequency bands). This way you can use the crossover filter in Metaplugin to create multiband versions of all possible VST plugins!



The GUI of the *Mid-Side matrix* consists of only one knob: when in “LR->MS” mode the plugin takes a stereo signal on input and delivers the mid and the side part of this signal on output. In “MS->LR” mode the reverse thing happens. This way, mid-side treatment for all kinds of plugins and even chains of plugins become possible.

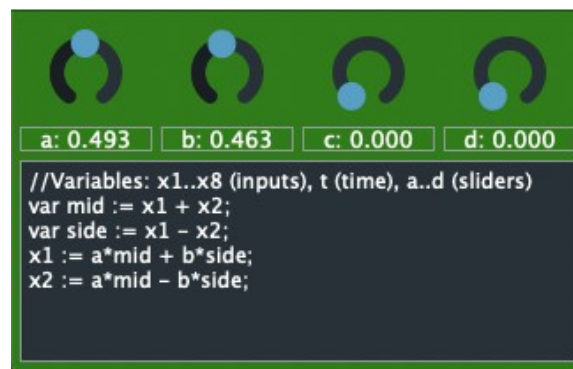


With *SendIt*, you can use up to 64 "channels" to send audio from one instance to another. This is somewhat of a legacy tool from back when Metaplugin was not yet supporting its own sidechain channels, but might still be useful for hosts that do not support sidechaining.



Say you want to stream the audio signal from track A into a Metaplugin instance which is loaded on track B. For this, put an instance of SendIt on track A, choose a free channel and use the "send" mode. Load another instance in Metaplugin on track B, using the same channel and "receive" mode. The audio signal from track A will now be streamed into the SendIt plugin in Metaplugin on track B and can be used in there in any way you like. You may or may not check the "PassThru" option in SendIt, which will route any audio going into it to its output. Please note that there's only one possible send/receive combination for each channel!

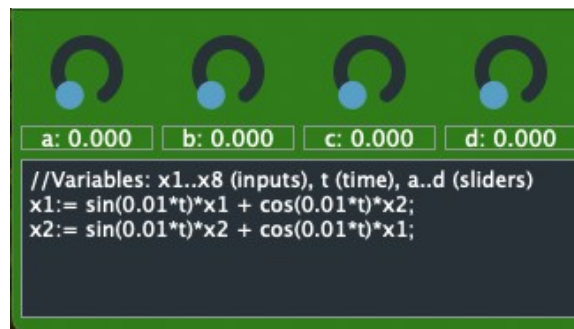
The *FunctionPlugin* is a very useful little tool that gives you full control over what is happening to the incoming audio samples by an on-the-fly parsable formula editor. That feature is based on the amazing ExprTk library by Arash Partow (<https://www.partow.net/about/index.html>). The FunctionPlugin has 4x2 inputs and a stereo output. The 8 input channels can be accessed by using the variables x1..x8. Furthermore, you have 4 independent variables a..d that can be controlled by 4 sliders. Finally, you have access to the current time t. Your input will continuously be compiled, and if the plugin will show a green background if it has parsed the text successfully, and a red background if there are some errors or if your input is simply not finished yet. Here are some examples:



This is a mid-side plugin (operating on the first 2 inputs). You can control the amount of mid and side signal with the parameter sliders "a" and "b". Note how you can define additional local variables by

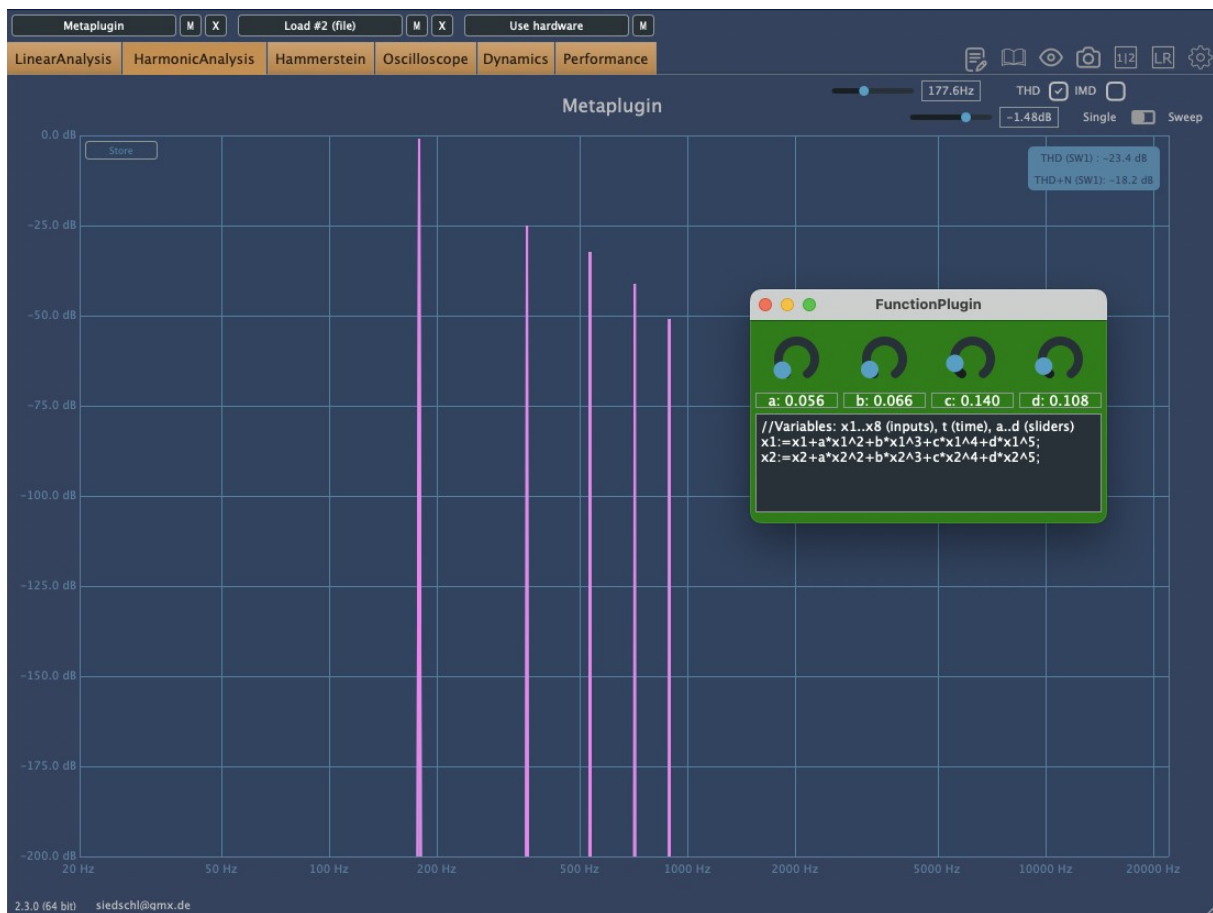
using the “var” keyword. Every assignment has to be done with “:=”, and every line has to end with “;”.

Here is another example:



This is a set of formulas that lets the left and right channel (again, this example is for the first two inputs only, but everything can easily be extended to the rest) with a certain frequency. Note the use of the value of the time parameter, “t”.

Finally, a way to quickly create your own higher harmonics:



Using the powers x^2 , x^3 , x^4 and x^5 , you can adjust the contributions of harmonics to the input signal with the 4 parameters a, b, c and d.

The possibilities here are enormous, of course... For further reference of what the formula parser can do, please refer to https://www.partow.net/programming/exp_rtk/index.html . **WARNING: it's easy to create unwanted noise bursts if you don't know what you are doing! For safety reasons, there is an output limit of +40dBfs, but even that can already be potentially harmful. So please think about what you are doing, and be careful with those exponentials and divisions by (nearly) zero!**

Plugin wrapping between different hosts

Apart from creating all sorts of plugin chains, Metaplugin can also be used as a wrapper/bridge between different plugin formats. For instance, you can load VST instruments in ProTools or Logic (by using the AAX or AU version of Metaplugin and loading VST plugins into it), or AU plugins in a host that only supports the VST format. Additionally, if you are on a newer Mac with an ARM processor, you can load plugins that are only available in Intel format when running Metaplugin in an Apple Silicon host. This is accomplished using the "Bridged Intel" formats that you will find as additional plugin format options.

Limitations

Please note that, for legal reasons, it is not possible to load AAX plugins into Metaplugin, which means that you cannot load ProTools plugins in, say, Cubase or Logic. Also, feedback loops are not supported at the moment. Other than that, go wild!

One more note for Mac users: when experiencing issues with the loading of older AU plugins, it usually helps to use the corresponding VST or VST3 versions. There are some things that have changed over the last few years in the AU standard (especially concerning the handling of GUI commands), which can occasionally lead to non-responsive GUIs in Metaplugin.

Demo restrictions

If you use the demo version of Metaplugin, the settings will not be saved when saving a project/song. Apart from that there are no other restrictions so you can fully explore the possibilities it offers to you.

Questions/feedback: support@ddmf.eu