DDMF Metaplugin: manual

Version 2.6.0

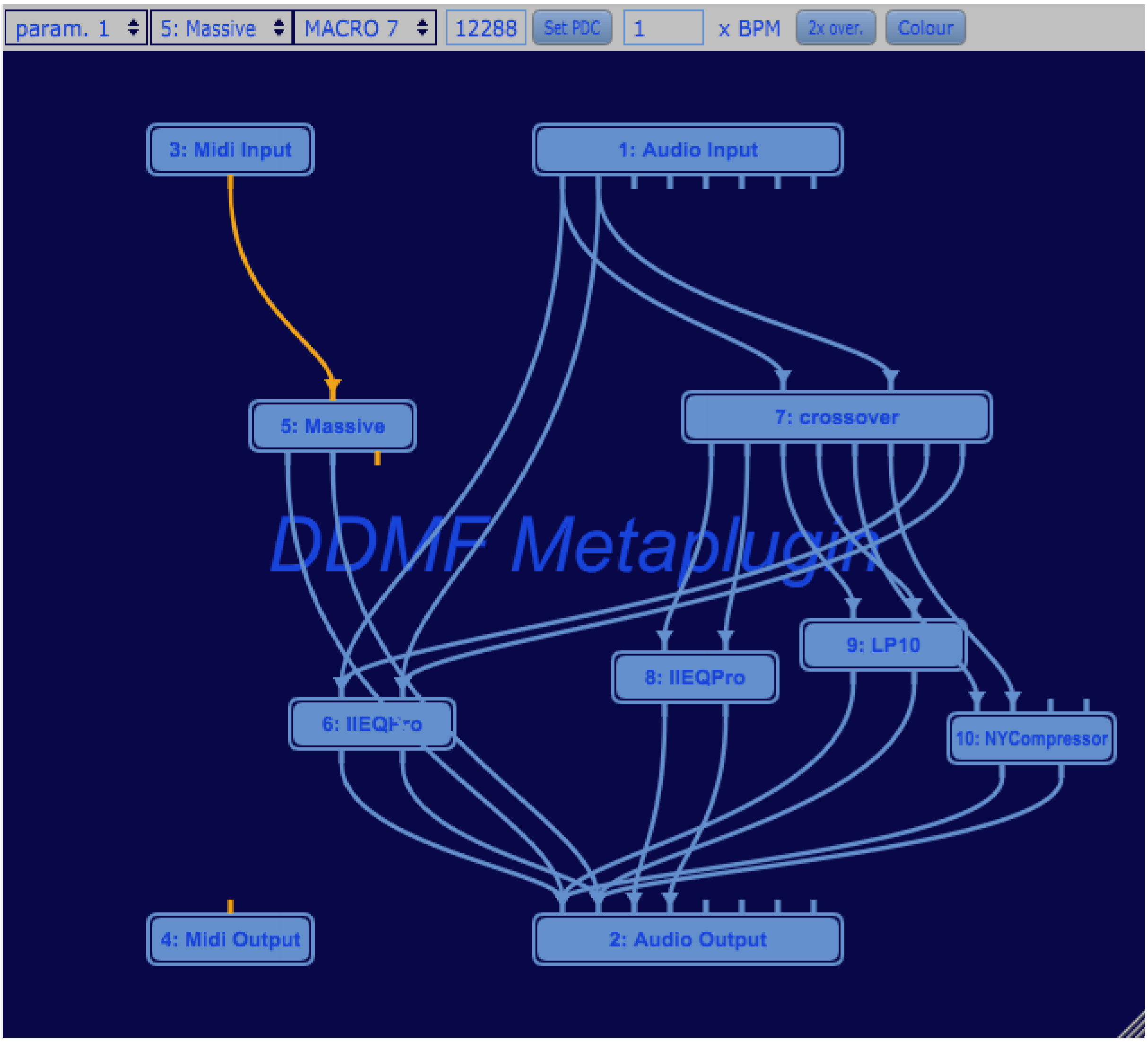
Thank you for using DDMF's Metaplugin! The following should help you to get the most out of this audio plugin. The possibilities are enormous but the principle is actually very simple. If, after reading, you still have questions about the usage of this software, don't hesitate to contact support@ddmf.eu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

Setup:

Metaplugin comes as a (Windows or Mac) installer package. The available plugin formats are VST, AU (Mac only), RTAS 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. For Mac, please note that even if you only want to use RTAS or AAX, you need to check either the VST or the AU box if you want to use the two helper plugins (Mid-Side and four band crossover) that come with Metaplugin, as those two are VST/AU plugins. 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/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. The audio output here consists of 6 channels, and the various signals can be distributed among them as you like.

But how are plugins loaded into Metaplugin in the first place? There are two ways to accomplish this: 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.

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.

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. 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 with 2x-oversampling. Oversampling refers to the process 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). To operate in 2x oversampling mode, just press the 2x-over-button which will then be shown in red. Please note that you'll have to delete and reload all plugins after that if you want the "new" samplerate to be transmitted to them.

Accessing plugin parameters

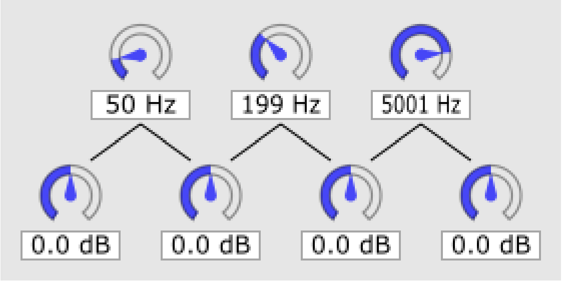
Metaplugin exposes 100 parameters to the host. These parameters can be mapped onto the parameters of loaded plugins using the three parameter mapping drop-down menus in the top row. For instance, if you have loaded an effect called SuperEQ which has an internal parameter called HighGain and you want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

Helper plugins

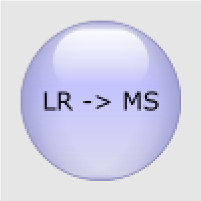
As an extra bonus, three DDMF plugins are included in the package that were specifically designed for what we think are typical usages of Metaplugin: in the Crossover plugin you find a flat-summing four band crossover filter using Linkwitz-

Riley filters. 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. One typical application would be to create a side-chain input in Metaplugin: 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!

☑ Send
● Receive
☑ PassThru
Channel 1

Plugin wrapping and 32/64 bit hosts and plugins: 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. One possible obstacle here can be caused by 64-bit-only hosts into which you'd like to load 32-bit-only-plugins. If you are running Metaplugin in such a 64 bit hosts, the user interface will display "Metaplugin64", which for you, the user, means that only 64 bit plugins can be loaded. Fortunately there are possibilities to "bridge" 32 bit plugins to the 64 bit world; DDMF recommends jBridge (https://jstuff.wordpress.com/jbridge/) to achieve this. This very helpful software takes 32 bit VST plugins and creates their 64 bit counterparts, which can then be loaded into Metaplugin64.

Limitations: Please note that, for legal reasons, it is not possbile to load RTAS or 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 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 of this effect.

Questions/feedback: support@ddmf.eu