Portal plugin

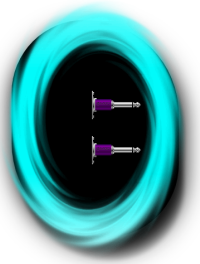
When used correctly, it will significantly reduce the CPU percentage used for long and heavy signal chains. **Expect a drop of 30 to 40% on your typical chains.**

The Portal plugin allows your MOD device to split a single, heavy audio processing chain across multiple CPU cores, allowing more load before maxing out, for the small price of 1 added audio cycle: a latency of 2.6ms. This latency is in many cases unnoticeable or negligible.

How to use the Portal Plugin?

In practice, this plugin only works if you use BOTH components of the portal:

the entry point: the “Sink” (yellow)



The exit: the Source (blue)

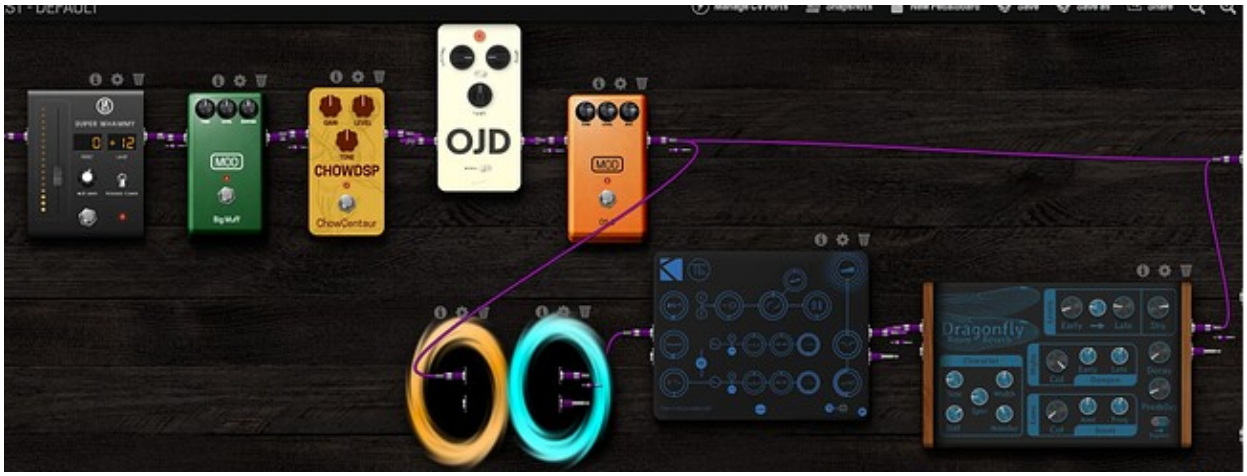
Although they appear as two separate plugins in the plugin list, you need to add both of them to your pedalboard to make effective use of the Portal.

Best practices

In practice, pedals are usually divided in 2 areas, with the portal split between the two. These best practices are a general recommendation, but results may vary, depending on your specific pedalboard. Use it as a boilerplate and experiment with placing it from there.

Example board

Notice how some overdrive and distortions are placed before the Sink (yellow).



Plugins best placed BEFORE the Portal:

- Pitch Shifters
- Octavers
- Compressors
- Overdrives
- Distortions
- Fuzz
- Amp Sims
- Amp profile plugins (NAM, AIDA-X, etc):

Plugins best placed AFTER the Portal:

- Heavy Cabinet IR Loaders
- Complex Synths & Ambient Textures
- Reverbs
- Delays

Avoid phasing

Never send one half of your audio signal through the Portal, and mix it back together with the un-Portaled signal. In other words, you can create loops before or after the

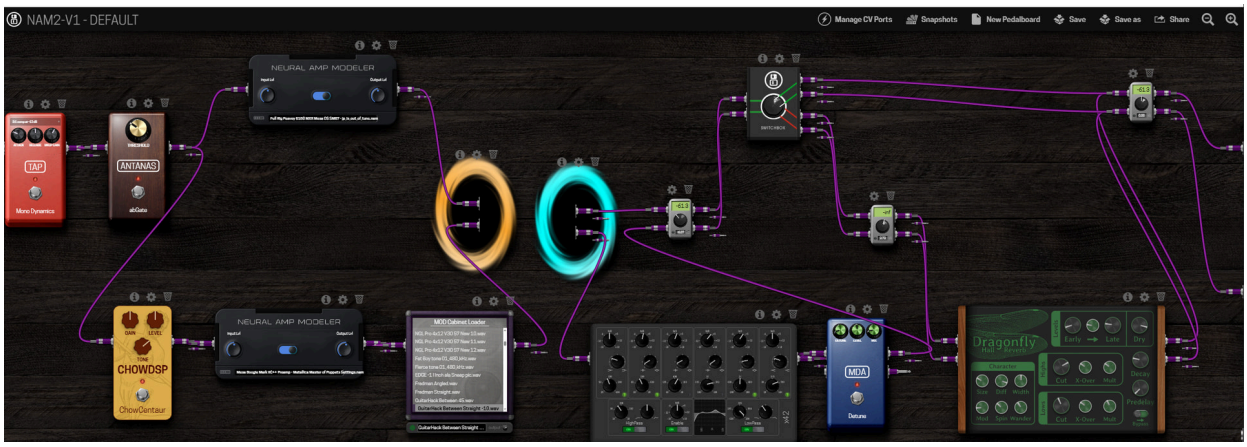
Portal, but never bridge over from the “before” side to the “after” side of the portal or vice versa.

Because the Portal adds a 2.6ms delay, mixing it directly back with the "instant" dry signal will often cause comb-filtering and phasing (a hollow, swooshing sound that ruins your tone).

If you want to mix dry and wet signals, do it *before* the Portal. Send the 100% wet signal through the Portal to your reverbs/delays, and route both chains entirely separately to the final physical outputs of the MOD device.

Example setup

This example splits a guitar signal into parallel paths with a different amp simulator on each path. In the end, 2 different flavours of the same guitar end up in the outputs. Notice that the signal gets split and merged several times before and after the Portal but never bridges it to make a blend of a “portaled” and an “unportaled” connection. Note: This particular example uses an IR loader before the portal, but the user chose to do this to keep the simulators and “time-based” sections divided by the Portal. This isn’t a functional rule; you are free to place and use the Portal as you see fit.



Experiment with your specific pedalboard.

The flexibility of the MOD devices allows you to elaborate and create large pedalboards. The success of this guide may vary based on your specific pedalboard. If the results disappoint, try shifting the placement to the left or right in your chain, while always keeping the phasing advice in mind

How does it work?

By default, plugins process audio in a series, like an assembly line, using only one CPU core.
Example: If you have four plugins, each plugin has to wait for the one before it. If one of those plugins requires a lot of processing power, the whole assembly line gets delayed.

By placing the portal in your chain, you force the audio path to split into two parallel paths. This allows the second half of your pedalboard to run on an entirely separate CPU core, effectively doubling the processing headroom for that specific chain. This process costs exactly 1 audio cycle of latency (about 2.6ms at a standard 128 buffer size).

So what visually happens in the pedalboard is slightly different than what happens in the background, but this is done to keep the interface uncluttered with complex routing visualisation.

PROs and CONs

The Portal has a big and obvious upside: creating more free processing power. This comes with the cost of an added latency, which, when used properly, is barely noticeable.

The Pros

- CPU Relief: By spreading the load across multiple cores, you can drastically reduce your main CPU load.
- Allows for larger pedalboards: You can now build massive chains with multiple heavy plugins that would normally crash a single-core setup.
- Host-Level Control: Because it is handled via a plugin rather than an automated background process, you retain manual control over where the audio routes and splits.

The Cons

- Introduces Latency: The "teleportation" process costs exactly 1 audio cycle of latency (about 2.6ms at a standard 128 buffer size). For most of us, a 2.6ms delay is unnoticeable.
- Risk of Phasing Issues: Because the added 2.6ms delay, if you split a signal using Portal and then try to merge it back together with the original dry signal later, the slight time difference can cause unpleasant comb-filtering or phasing.

- Strict Limit: You can only load one Sink and one Source per pedalboard. You cannot create endless parallel setups.
- Manual Setup Required: The system doesn't automatically know which plugins are hogging your CPU. You have to figure out where the bottleneck is and manually place the Portal plugins in the smartest location in your chain.