

PYRITE



Table of Contents

Table of Contents	1
Introduction	2
Main features	2
Installation	2
Signal Perspective toolbar	3
➤ Note about resizing:	3
➤ Note about preset files:	3
➤ Note about preset names:	3
Using Pyrite	4
Acknowledgments	5
Disclaimer	5

Introduction

Pyrite is a virtual-analog preamplifier emulation, it features an input transformer, an EQ, and a peculiar compressor, plus some old-school meters.

It really shines on drum sounds, but can also be used on other types of tracks.

The **transformer** can be used to saturate the very low end, and also as a hi-pass filter on the input stage. The **saturation** can be set to a soft or hard knee, with two screws to tune the modelling of the original device. There is also a “blend” control that allows you to mix the unsaturated signal back in. The **dynamics processor** is a soft-ish-knee compressor that focuses more on bringing out the quiet parts than on taming the peaks (at least at moderate values of the Amount knob); it sounds a bit like a parallel compression algorithm. The **EQs** are a faithful reproduction of the original unit’s low-cut and air filters, but with a wider and more continuous range, plus a low-end lift filter modeled on the (linear) behavior of an EMI REDD console bass lift function.

Main features

The following features are distinctive of Pyrite:

- 192kHz / 64bit internal oversampling
- Tunable transformer
- Analog emulated saturation
- Unique compression algorithm
- Hi-pass, Hi-shelf, and Low-shelf EQ
- A “comments” area for making notes that are saved with the preset or plug-in instance state.

Installation

The Windows version comes with an installer application that copies the presets and the plugin files in the correct folders. Under macOS you need to move the plugin files and the presets files and folders manually to the appropriate folder:

macOS VST3 folder:	~/Library/Audio/Plug-Ins/VST3
macOS AU folder:	~/Library/Audio/Plug-Ins/Components
Windows VST3 folder:	C:\Program Files\Common Files\VST3
Presets folder:	<i>User Documents Folder</i> /Signal Perspective/Pyrite

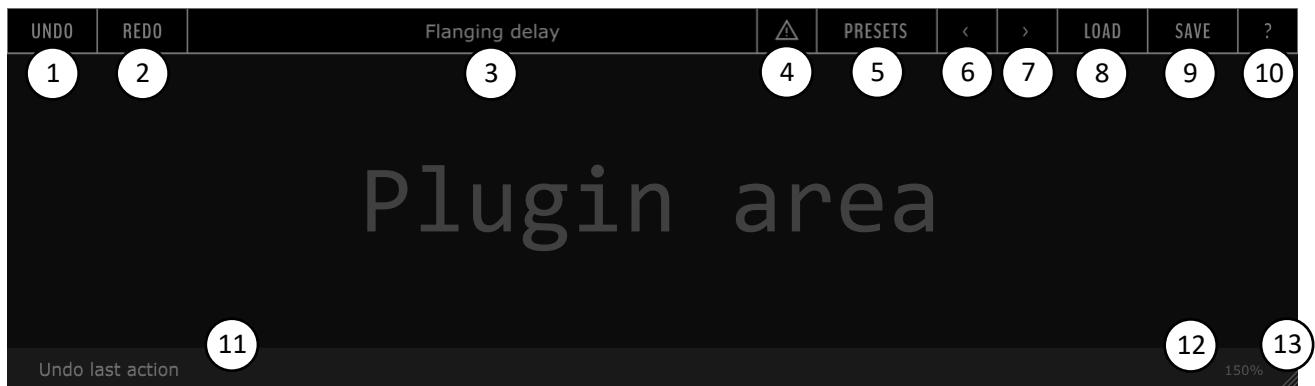
Please note that the plugin is not digitally signed: if you have trouble running the plugin under macOS, try typing the following in the OS terminal or visit the product page for further instructions:

```
sudo xattr -d -r com.apple.quarantine ~/Library/Audio/Plug-Ins/VST3/Pyrite.vst3  
sudo xattr -d -r com.apple.quarantine ~/Library/Audio/Plug-Ins/Components/Pyrite.component
```

The standalone application is portable, meaning that you can bring the executable wherever you want.

Signal Perspective toolbar

Most Signal Perspective plugins share a toolbar that lets you navigate presets and handle some common features, such as a tooltip system that lets you learn how to use the plugin without the need for a manual.



- | | |
|------------------------------|--|
| 1. Undo button | 8. Load specific preset file |
| 2. Redo button | 9. Save preset file |
| 3. Preset name | 10. Open this manual |
| 4. Panic button (if present) | 11. Tooltip area / Editable preset comment |
| 5. Preset menu | 12. Zoom factor (reset to 100% if clicked) |
| 6. Load previous preset | 13. Resizing handle |
| 7. Load next preset | |

➤ Note about resizing:

Each time the plugin is resized it saves the current scale in a file so that new instances can already be scaled as desired. If you inadvertently get the plugin larger than the screen size (and therefore you can no longer resize the window) you can force it to open to 100% by deleting the settings file, which under Windows is:

`%APPDATA%\Pyrite\Pyrite.settings`

and under macOS:

`~/Library/Application Support/Pyrite.settings`

➤ Note about preset files:

In order to let the plugin/application see the preset files and show them in the preset menu, be sure to place them in the **user documents folder** under the `/Signal Perspective/Pyrite` subfolder, without altering the provided tree structure. The drop-down menu shows all the folders present as submenus, populated by the XML files contained in them. No further subdirectories are shown (i.e. there is no recursive search). The drop-down menu is repopulated after each preset saving and loading operation, and upon reopening the interface.

➤ Note about preset names:

The preset name is stored inside the preset file and is set to be the same as the file name when saving a preset, but if you rename the file, the actual preset name will remain unchanged.

Using Pyrite

The plugin is organized into 4 sections:



1. **Saturation:** From here it is possible to define the non-linear behavior of the plugin:
 - **-oo- button** turns the transformer on and off, affecting very low frequencies;
 - **Imp** controls the input impedance (when the transformer is turned on);
 - **Tuning** trim changes the size of the transformer, resulting in a different frequency range where distortion occurs and acting as a high-pass filter below that frequency;
 - **Saturation** controls the amount of preamp saturation;
 - **Pow cap** trim changes the distortion shape. Harsh on the left and more dynamic on the right;
 - **Bias cap** trim controls the DC build-up speed. Turn it to the left if the signal is too distorted or muted when a quiet program follows a very loud sound;
 - **Blend** is a dry-wet control that blends the distorted signal with the input signal;
 - **Shape** switches between hard-knee distortion or soft-knee distortion.
2. **EQ:** This is where you define the linear behavior of the plugin (applied after the compressor):
 - **HPF** is a Hi-pass filter;
 - **Air** is a Hi-shelf filter;
 - **Lift** is a very-low-shelf filter.
3. **Compression:** This is the compression stage of the plugin (applied before EQing):
 - **Thresh** controls the compressor threshold;
 - **Amount** controls the amount of compression;
 - **Atk** lets you choose between fast – medium – slow attack times;
 - **Rel** lets you choose between fast – medium – slow release times;
 - **Ch** switch between independent L/R compression or same compression on both channels.
4. **Output:** This is the output stage of the plugin:
 - **Output** controls the output signal level;
 - **BYP** just bypasses the whole unit.

Other details about each parameter are available as tooltips in the plugin.

In order to use it properly, you need to understand the topology of the included processors (which differs from what you might assume based on the interface):

Transformer -> Saturation -> Compression -> EQing -> Output trimming

Acknowledgments

Many thanks to the Pyrite Beta testers:

- Matteo Bolpagni
- Marco Tiraboschi
- Rob Connelly

Disclaimer

This software was developed for non-commercial purposes by Giorgio Presti, we decline responsibility for any malfunctions that may cause loss of data or other inconveniences.

www.signalperspective.com

info@signalperspective.com