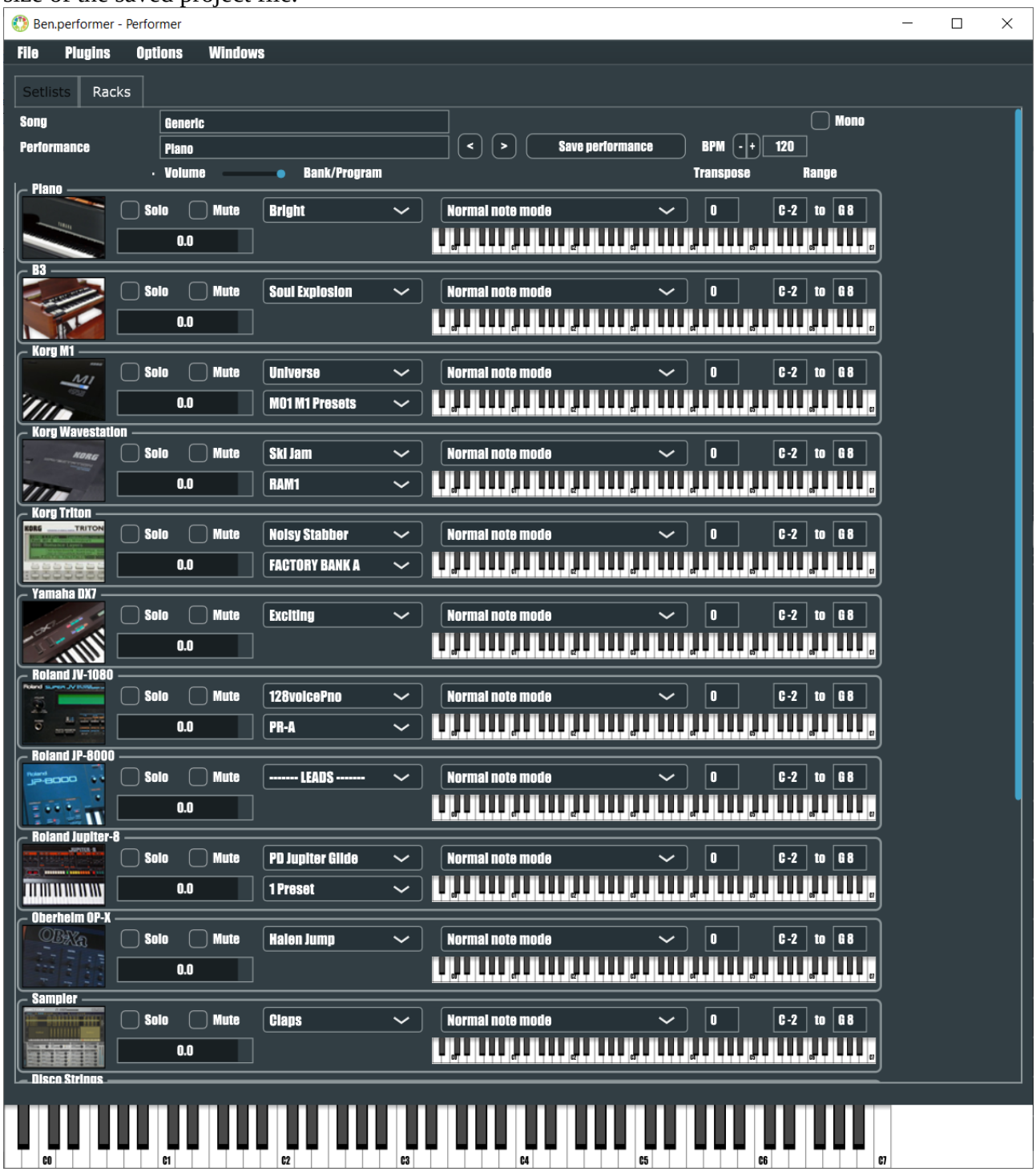
Performer

Introduction

Performer is a highly customised audio plugin host for MacOS and Windows. It tries to use the simple concept of having physical racks which you would change the sound by sending “program change” midi events. A lot of modern day plugins don’t do things this way anymore but where possible it is used. One of the advantages of this is program changes are usually a lot faster than sending large state data which is great when you are switching between songs. This also reduces the size of the saved project file.



The project files are saved in xml text removing the problem of unreadable binary files.

It also has support for the Arturia Keylab where it will send sysex events to the Keylab to show patch names on the LCD screen.

Getting started

First thing to do is setup your audio system. Menu / Options / “Change the Audio Device Settings”

If you want the Keylab LCD song display then put the Keylab in your “MIDI Output”

Next you need to tell Performer what plugins you have.
Menu / “Edit the List of Available Plug-ins”
Then click Options / “Scan for new or updated VST plug-ins”
Enter in all the paths where your plugins live and then click Scan.

You can now add your plugins to your rack.
Menu / Plugins / “Create Plug-in”

Racks

Once you've added a plugin you have various settings on it. A volume in dB. A midi range to limit when it plays. A midi transpose. A program dropdown which will be midi program changes. A mute toggle for songs you don't want to use that plugin. A special dropdown for midi effects such as appregiator or simply double octave. Any changes you do are temporary until you click "Save performance" (except global settings like rack order or global state)

If you click on the rack image it will show the plugins main window. Right clicking on this image will present a few other operations which you can use to order your racks, delete them and rename. Here is also where you manage plugin state. As mentioned earlier, if the plugin supports program changes then you might not even have to use state. Examples of this are Korg M1 / Korg Wavestation / Korg Triton / Roland JV-1080 / Roland Jupiter-8 / TruePianos. The Roland plugins have a bug where they don't respond to program changes but Performer fixes this.

The image used for the plugin is just a png file named the same as the rack.

For plugins that don't handle getting program names, a text file can be used named the same as the rack. An example of this is "Piano.txt" which could list the Diamond collection for TruePianos.

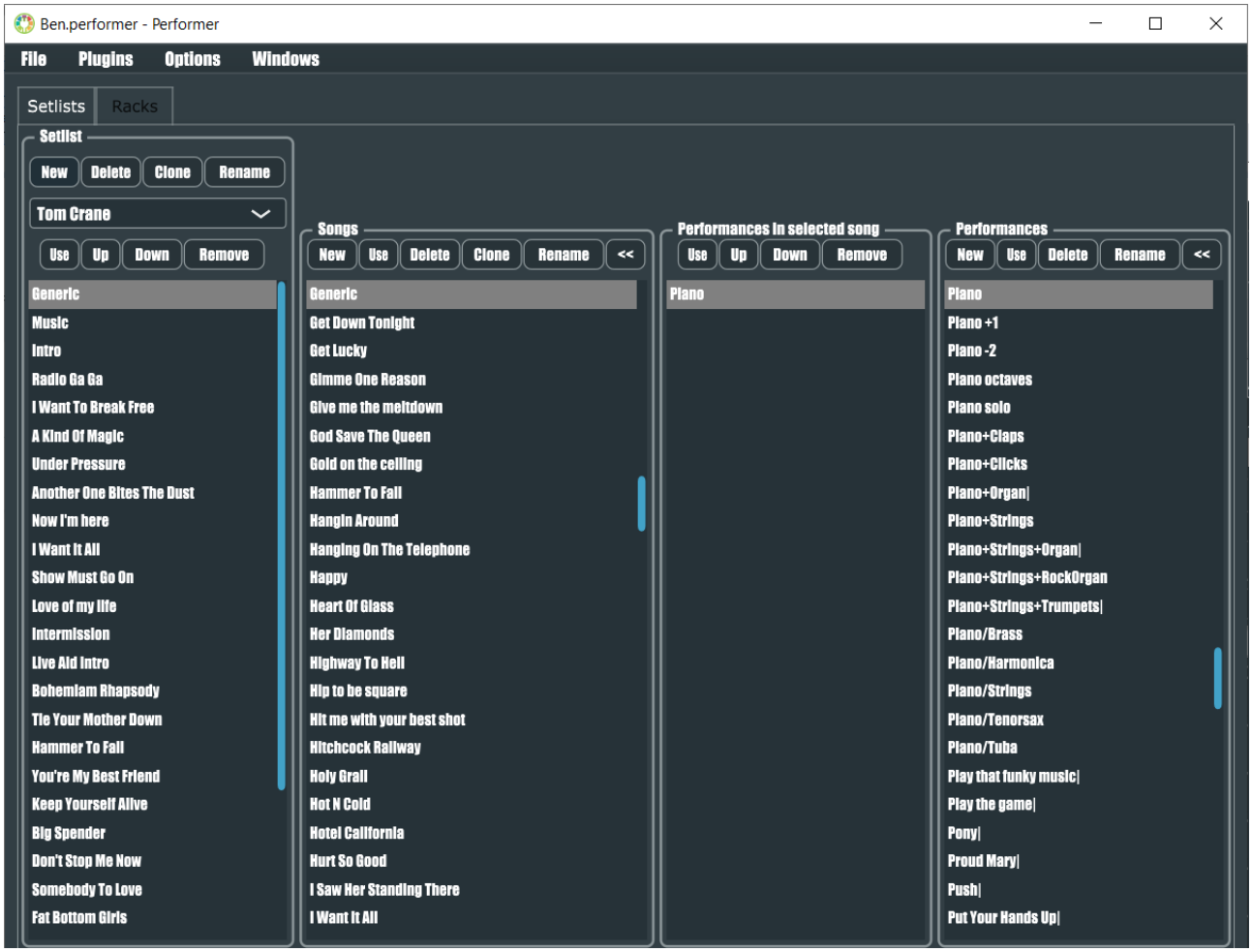
For plugins that use banks, a text file named "Banknames.txt" in a folder "PresetNames" and then the same of the **plugin**. Examples of this are the Korg and Roland plugins.

There are two types of state a plugin has. One is global. It gets sent once and then program changes take it from here. An example of this could be Native Instruments FM8. Once setup with a program change list for the legacy FM7, it's global state can be saved. Another example is Native Instruments B4 II. By default the modulation wheel isn't set to rotary speed. By setting this up once and saving as global, the wheel is always working.

Of course sometimes you still want full state sent for each song. There is a state override for this that can be stored per performance.

Setlists

The racks represent the setup for one performance. A song may be made up of multiple performances. A song will also be in a setlist and you could have multiple setlists. All this information is handled on the Setlists tab



Advantages over existing software

- No complex wiring
- Use program changes / presets (including fix for Roland plugins) or full plugin state
- Wav streamer / Soundfont player / Guitar strummer included
- Full setlist / song organizer
- Human readable future proof project files
- Interfaces with Arturia hardware
- Supports MacOS and Windows
- Compatible with jBridge to use legacy 32bit plugins