

# Loomer String User Manual

# Contents

<b>Licences</b>	<b>1</b>
End User Licence Agreement . . . . .	1
SIL Open Font License . . . . .	2
<b>Requirements</b>	<b>5</b>
macOS Requirements . . . . .	5
Windows Requirements . . . . .	5
Linux Requirements . . . . .	5
<b>String Overview</b>	<b>6</b>
What is String? . . . . .	6
Plug-in vs Standalone . . . . .	8
String Formats . . . . .	8
String FX . . . . .	8
Supported Channel Configurations . . . . .	8
<b>Installation and Registration</b>	<b>10</b>
macOS Installation . . . . .	10
Windows Installation . . . . .	10
Linux Installation . . . . .	11
Product Registration . . . . .	11
<b>Using String</b>	<b>13</b>
LAYER 1 and LAYER 2 . . . . .	14
SPLITS . . . . .	15
MODULATION . . . . .	15
EFFECTS . . . . .	16
MAIN . . . . .	16
String FX Interface . . . . .	17
Control Types . . . . .	17
Rotary Controls . . . . .	17
Button Controls . . . . .	18
Slider Controls . . . . .	18
Setting the Interface Size . . . . .	18

<b>Preferences</b>	<b>19</b>
Configure Audio and MIDI . . . . .	19
Configure Appearance . . . . .	20
Theme . . . . .	21
Configure Key Bindings . . . . .	21
Configure Default Preset Author . . . . .	21
Configure MIDI Learn . . . . .	22
<b>MIDI Control</b>	<b>23</b>
MIDI Controllers . . . . .	23
Display Current MIDI Controller Mappings . . . . .	23
Assign MIDI Controllers to Controls . . . . .	24
Remove MIDI Controller Mappings From Controls . . . . .	24
MIDI Program Change Lists . . . . .	25
<b>Preset Management</b>	<b>27</b>
Parameters and Presets . . . . .	27
Navigation Controls . . . . .	27
Creating a New Preset . . . . .	28
Comparing a Changed Preset to the Original Preset . . . . .	28
Opening Presets . . . . .	28
Saving Presets . . . . .	28
The Preset Browser . . . . .	29
Preset Hot-swap . . . . .	30
Preset Meta Data . . . . .	30
Searching for Presets . . . . .	30
Categories . . . . .	30
Preset Formats . . . . .	30
Copying and Moving Presets . . . . .	31
The Default Preset . . . . .	31
Restoring Deleted Factory Presets . . . . .	31
User Folder Management . . . . .	31
Favourites . . . . .	31
<b>Status display</b>	<b>32</b>
Tempo . . . . .	32
Polyphony . . . . .	32
CPU Usage Monitor . . . . .	33
MIDI Input Monitor . . . . .	33
<b>Contact Details</b>	<b>34</b>
Online . . . . .	34
Email . . . . .	34

# Licences

## End User Licence Agreement

### Software Licence Agreement for String

1. **GRANT OF LICENCE:** Subject to the terms below, Loomer hereby grants you a non-exclusive, non-transferable licence to install and to use String ('Software'). Under this licence, you may: (i) install and use the Software for your personal, internal use (ii) copy the Software for back-up or archival purposes; (iii) use an unregistered or evaluation version of the Software for evaluation purposes only. (iv) You may not distribute the software to others without first obtaining the required licences, where applicable. Whether you are licensing the Software as an individual or on behalf of an entity, you may not: (i) reverse engineer, decompile, or disassemble the Software or attempt to discover the source code; (ii) modify, or create derivative works based upon, the Software in whole or in part without the express written consent of Loomer; (iii) distribute copies of the Software; (iv) remove any proprietary notices or labels on the Software; (v) resell, lease, rent, transfer, sublicense, or otherwise transfer rights to the Software; By installing, copying, or using the Software, you (either on behalf of yourself as an individual or on behalf of an entity as its authorized representative) agree to all the terms of this End User Licence Agreement ('Agreement') regarding your use of the software. If you do not agree with all the terms of this Agreement, you may not use the Software.
2. **STRING:** You acknowledge that no title to the intellectual property in the Software is transferred to you. Title, ownership, rights, and intellectual property rights in and to the Software shall remain that of Loomer . The Software is protected by copyright laws and international treaty provisions.
3. **DISCLAIMER OF WARRANTY:** You agree that Loomer has made no express warranties, oral or written, to you regarding the product and that the products are being provided to you 'as is' without warranty of any kind. Loomer disclaims any and all other warranties, whether expressed, implied, or statutory. Loomer shall not be liable for indirect, incident, special, cover, reliance, or consequential damages resulting from the user of this product.
4. **LIMITATION OF LIABILITY:** You use this program solely at your own risk. In no event shall Loomer be liable to you for any damage, including but not limited to any loss, or

other incidental, indirect or consequential damages of any kind arising out of the use of the software, even if Loomer has been advised of the possibility of such damages. In no event will Loomer be liable for any claim, whether in contract, tort, or any other theory of liability, exceed the cost of the software. This limitation shall apply to claims of personal injury to the extent permitted by law.

5. **TERMINATION:** This Agreement shall terminate automatically if you fail to comply with the limitations described in this Agreement. No notice shall be required to effectuate such termination. Upon termination, you must remove and destroy all copies of the Software.
6. **MISCELLANEOUS:** In the event of invalidity of any provision of this Agreement, the parties agree that such invalidity shall not affect the validity of the remaining portions of this Agreement. You agree that this is the entire agreement between you and Loomer, which supersedes any prior agreement, whether written or oral, and all other communications between Loomer and you relating to the subject matter of this Agreement.

Reservation of rights.

'VST' is a trademark of Steinberg Soft - und Hardware GmbH. All other trademarks are the property of their respective owners. All rights not expressly granted in this Agreement are reserved by Loomer.

## SIL Open Font License

This Font Software is licensed under the SIL Open Font License, Version 1.1. This license is copied below, and is also available with a FAQ at: <http://scripts.sil.org/OFL>

Lato.ttf: Copyright (c) 2010-2015, Łukasz Dziedzic (dziedzic@typoland.com), with Reserved Font Name Lato. This Font Software is licensed under the SIL Open Font License, Version 1.1.

Oswald: Copyright (c) 2012, Vernon Adams (vern@newtypography.co.uk), with Reserved Font Name Oswald. This Font Software is licensed under the SIL Open Font License, Version 1.1.

Poppins: (c) 2014-2017 Indian Type Foundry (info@indiantypefoundry.com), with Reserved Font Name 'Poppins'. This Font Software is licensed under the SIL Open Font License, Version 1.1.

---

SIL OPEN FONT LICENSE Version 1.1 - 26 February 2007

---

**PREAMBLE** The goals of the Open Font License (OFL) are to stimulate worldwide development of collaborative font projects, to support the font creation efforts of academic and linguistic communities, and to provide a free and open framework in which fonts may be shared and improved in partnership with others.

The OFL allows the licensed fonts to be used, studied, modified and redistributed freely as long as they are not sold by themselves. The fonts, including any derivative works, can be bundled, embedded, redistributed and/or sold with any software provided that any reserved names are not used by derivative works. The fonts and derivatives, however, cannot be released under any other type of license. The requirement for fonts to remain under this license does not apply to any document created using the fonts or their derivatives.

**DEFINITIONS** “Font Software” refers to the set of files released by the Copyright Holder(s) under this license and clearly marked as such. This may include source files, build scripts and documentation.

“Reserved Font Name” refers to any names specified as such after the copyright statement(s).

“Original Version” refers to the collection of Font Software components as distributed by the Copyright Holder(s).

“Modified Version” refers to any derivative made by adding to, deleting, or substituting – in part or in whole – any of the components of the Original Version, by changing formats or by porting the Font Software to a new environment.

“Author” refers to any designer, engineer, programmer, technical writer or other person who contributed to the Font Software.

**PERMISSION & CONDITIONS** Permission is hereby granted, free of charge, to any person obtaining a copy of the Font Software, to use, study, copy, merge, embed, modify, redistribute, and sell modified and unmodified copies of the Font Software, subject to the following conditions:

- 1) Neither the Font Software nor any of its individual components, in Original or Modified Versions, may be sold by itself.
- 2) Original or Modified Versions of the Font Software may be bundled, redistributed and/or sold with any software, provided that each copy contains the above copyright notice and this license. These can be included either as stand-alone text files, human-readable headers or in the appropriate machine-readable metadata fields within text or binary files as long as those fields can be easily viewed by the user.
- 3) No Modified Version of the Font Software may use the Reserved Font Name(s) unless explicit written permission is granted by the corresponding Copyright Holder. This restriction only applies to the primary font name as presented to the users.
- 4) The name(s) of the Copyright Holder(s) or the Author(s) of the Font Software shall not be used to promote, endorse or advertise any Modified Version, except to acknowledge the contribution(s) of the Copyright Holder(s) and the Author(s) or with their explicit written permission.
- 5) The Font Software, modified or unmodified, in part or in whole, must be distributed entirely under this license, and must not be distributed under any other license. The

requirement for fonts to remain under this license does not apply to any document created using the Font Software.

**TERMINATION** This license becomes null and void if any of the above conditions are not met.

**DISCLAIMER** THE FONT SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF COPYRIGHT, PATENT, TRADEMARK, OR OTHER RIGHT. IN NO EVENT SHALL THE COPYRIGHT HOLDER BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, INCLUDING ANY GENERAL, SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF THE USE OR INABILITY TO USE THE FONT SOFTWARE OR FROM OTHER DEALINGS IN THE FONT SOFTWARE.

# Requirements

## macOS Requirements

- Requires macOS v. 10.7 or later, or macOS 11
- x86 based processor 1.0Ghz, or above, with SSE, or M1 with Rosetta 2
- 512 MB RAM

## Windows Requirements

- Windows Vista or above
- x86 based processor 1.0Ghz, or above, with SSE
- 512 MB RAM

## Linux Requirements

- x86 32-bit or or 64-bit based processor 1.0Ghz, or above, with SSE
- 512 MB RAM

The following shared object libraries are required:

- libfreetype
- libasound
- libXinerama



# String Overview

## What is String?

String is a virtual instrument that emulates the beautiful sound of the polyphonic string synthesizer.

String's authentic tone comes from its accurate emulation of the divide-down oscillator technology behind the lush ensemble synths of the 70s.

String has two individual sound generating Layers, each housing three phase-locked oscillator stops at 16', 8' and 4' ranges for layer 1, and 8', 4', and 2' ranges for layer 2. The relative volumes of each stop can be finely tuned to produce an enormous range of timbres. Both layers have controllable note split points. Layers can be placed so that each is controlled by a specific range on the keyboard, or even overlapped and stacked together. Each layer has its own voicing mode from a choice of monophonic with last, low, and high note priorities, paraphonic, and polyphonic. In polyphonic mode each voice has its own attack/release amplitude modulation envelope. In paraphonic mode all voices within a layer share a single monophonic attack/release amplitude envelope. The monophonic modes produce a single voice, which can be either the lowest, highest, or last note pressed.

Layers can be further shaped using the monophonic filter section. This section has several filter types; a highpass / lowpass pair of filters in series, an analogue resonant filter in either lowpass, bandpass, or highpass modes with various modulation options (including channel aftertouch), and a 9-Band EQ. The filter section can also be switched off altogether.

Both layers can be individually routed into the effects section. Layer 1 can also optionally be routed directly into layer 2's filter to produce serial filter tones.

String's chorus section is modeled on the bucket-brigade device that characterized the sound of the classic ensemble synths. Select from two different chorus algorithms for either an authentic vintage ensemble timbre, or a more modern, denser sound. A built-in stereo phaser can be used to add movement, depth, and interest to String timbres. The effects section is rounded off with two mono bucket-brigade device delay units with independent time and feedback settings, which can be tempo synced, and linked together into a single stereo delay.

All controls can be easily mapped to a hardware controller using the simplified MIDI Learn feature.

With optimised audio algorithms that won't overload your processor, rock-solid stability, and easy integration with any MIDI hardware controller, String is ideal for live work. With a simple and intuitive interface, flexible bank and program management, and total parameter automation, String fits right into your studio environment.

## Plug-in vs Standalone

Depending upon your individual studio setup, String can either be used as a plug-in component of a compatible host application, or as a standalone application requiring no other dependencies. Generally, by using the host application's routing flexibility, running as a plug-in allows easier integration with existing plug-in effects and instruments. If this integration is not required, the standalone version, without the burden of the host, performs very slightly better.

Presets and MIDI mapping assignments are identical in both standalone and plug-in versions. This means any sounds created in one format can be opened in the other.

## String Formats

For macOS, the following formats are provided:

- Standalone application (64-bit)
- Audio Unit (AU) plug-in (32-bit and 64-bit)
- VST2 plug-in (32-bit and 64-bit)
- VST3 plug-in (64-bit)
- AAX plug-in (64-bit)

These formats are available for Windows PC:

- Standalone application (64-bit)
- VST2 plug-in (32-bit and 64-bit)
- VST3 plug-in (64-bit)
- AAX plug-in (64-bit)

These formats are available natively for Linux:

- Standalone application (32-bit and 64-bit)
- VST2 plug-in (32-bit and 64-bit)
- VST3 plug-in (64-bit)

## String FX

External instruments can also be processed using String's beautiful effects section with the separate StringFX plug-in.

## Supported Channel Configurations

For maximum flexibility, String can be used in one of several possible input and output channel configurations. String is a true stereo effect: left and right channel separation is preserved. The following configurations are possible:

- mono to mono (1 input, 1 output)
- mono to stereo (1 input, 2 outputs)
- stereo to stereo (2 inputs, 2 outputs)

Note that some host applications may not support one or more of these configurations; consult your host's documentation for further details.

# Installation and Registration

## macOS Installation

1. Start the installation process by double-clicking the String installer application.
2. Read the introduction screen dialog text. Press *Continue* when you are ready to proceed.
3. Read the licence agreement. Click *Continue*, then *Agree* to confirm you have read and agreed to the licence terms.
4. Select a destination volume into which to install String, and then click *Continue*.
5. From the list of installable components, uncheck any formats that you do not wish to install. Click *Continue* to proceed.
6. To complete the installation, press *Install*.

The components are, by default, installed into the following directories:

- VST2 plug-in into */Library/Audio/Plug-Ins/VST*
- VST3 plug-in into */Library/Audio/Plug-Ins/VST3*
- Audio Unit plug-in into */Library/Audio/Plug-Ins/Components*
- AAX plug-in into */Library/Application Support/Avid/Audio/Plug-Ins*
- Standalone application into */Applications*
- Documentation into */Library/Documentation/Loomer*

## Windows Installation

1. Start the installation process by double-clicking the String installer application.
2. Read the welcome screen dialog text. Press *Next* when you are ready to continue.
3. Read the licence agreement. Click *I Agree* to confirm that you have read and agreed to it.
4. From the list of installable components, uncheck any formats that you do not wish to install. Click *Next* to continue.
5. If you have chosen to include the VST plug-in format, select the directory into which it will install. Click *Next* to continue.
6. Select a directory into which the common files will install. The user documentation and the Standalone application (if selected in the the list of components), will be placed in this location. To complete the installation, press *Install*.

Unless any other directories are selected during the installation, the components are installed into the following directories:

- VST2 plug-in into the VST plug-ins folder, which is, unless configured otherwise, *C:\Program Files\Steinberg\VSTPlugins*
- VST3 plug-in into the VST3 plug-ins folder *C:\Program Files\Common\VST3*
- AAX plug-in into *C:\Program Files\Common Files\Avid\Audio\Plug-Ins*
- Standalone application, documentation and default Bank into *C:\Program Files\Loomer\String*.

## Linux Installation

String for Linux is distributed as a standard Linux tarball. To install, extract the package using any unarchiving utility; the following command-line invocation will suffice:

```
tar xzvf String.tar.gz
```

If you have a Linux VST host and have configured your VST Path, the VST plug-in can be moved into the required directory with the following commands:

```
cd String
mv StringVST.so $VST_PATH
```

Without the VST Path configured, you will need to ensure that your host uses the String directory when searching for VSTs; consult your host's documentation for information on how to do this.

String requires the following shared object libraries to be installed:

- libfreetype
- libasound
- libXinerama

Without these libraries installed, the application will not launch. Install the libraries using your distribution's particular package management system, such as apt-get; consult your Linux distribution documentation for more information.

## Product Registration

Having installed String, it will run in evaluation mode with the following limitations:

- User created presets can be saved, but not loaded.
- Output will be periodically interrupted, approximately every minute, by a short moment of silence.
- Audio will be stopped altogether after approximately 30 minutes of continuous use. At this point, Shift2 must be restarted.

These limitations can be removed by purchasing a licence and registering your licence information into the product. Note that registering String on a particular machine will remove the evaluation limitations for all String plug-in formats installed on this machine; you don't need to individually register all plug-in formats. Licence information will be in the form of a user name and licence key.

To purchase a String licence:

1. Launch String. This can be either the standalone application, or as a plug-in.
2. From the *options* menu (the cog button on the top-right of the interface), choose *Purchase String Licence*. This will launch the shop website, from where a String licence can be purchased by clicking *Add To Cart*, followed by *Checkout*. Follow the on-screen prompts to enter your payment details and make a secure purchase.
3. Once you have purchased String, you will receive your licence key information via email. It is important to keep this information safe because you will require the licence key if you ever need to reinstall and re-register String. If you have lost your licence details, please [contact us](#). Licence keys are sent immediately upon purchase: if you haven't received yours within minutes of a purchase, either check your spam folder or [contact us](#).

To register your copy of String:

1. Choose *Enter String Licence* from the *options* menu.
2. Enter your name in the *Name* field. This must match exactly the name that was used to purchase your String licence, including capitalization of letters.
3. Enter your licence key in the *Licence Key* field. This must match exactly the licence that you received when you purchased String, including any hyphens. We recommend using *copy* and *paste* to ensure that the licence key is entered exactly as specified in your registration email.
4. Press *OK* to confirm that your name and corresponding licence key have been entered. You will be informed that the application must be restarted in order for registration changes to take effect. If you are currently running the standalone application, close and relaunch it. If String is currently running as a plug-in, remove it from the host and add a new instance of it.
5. String should now be registered. If the label on the bottom of the interface still reads *Unregistered Evaluation*, you must have entered the licence information incorrectly. Go back to step 1, paying careful attention that the name and licence key entered match exactly the licence information in the registration email.

If you are having problems registering your installation of String, please consult the detailed step-by-step instructions at [www.loomer.co.uk/support.htm](http://www.loomer.co.uk/support.htm).

# Using String

String can either be used a standalone application, or as a plug-in. When used as a plug-in effect, String is placed directly in a host instrument channel. Consult your host's documentation to find out how to add a plug-in instrument.

String comes as two separate plug-ins: String, and StringFX. The FX versions are shipped as separate plugins, and identified with the \_FX prefix. Add an instance of String to use the product as a powerful ensemble synthesizer. Add an instance of StringFX to use the product as an effects processor of other external instruments.

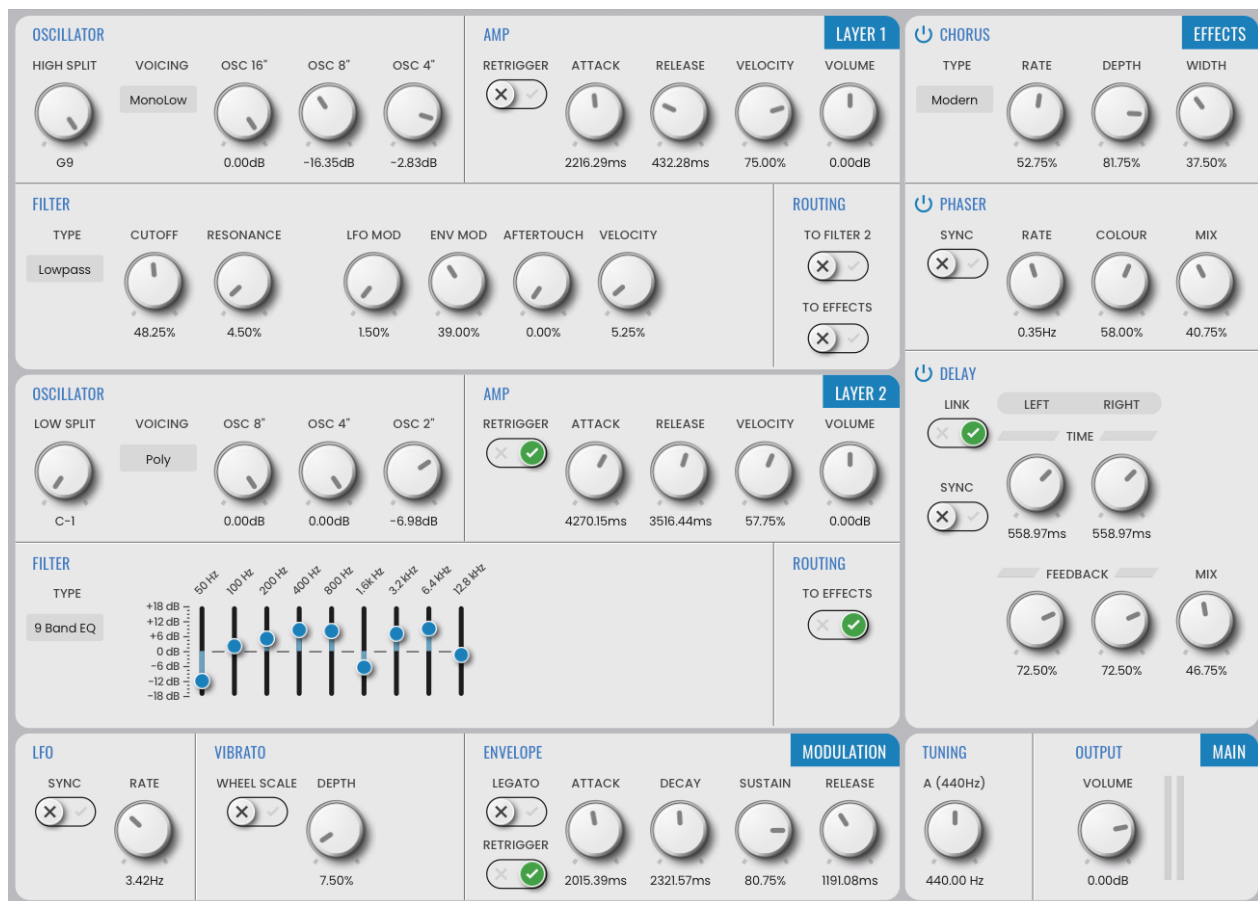


Figure 1: String's Interface



## LAYER 1 and LAYER 2

The *VOICING* parameter selects the number of voices that the layer will produce, and which note has priority. When only a single layer is required, selecting *Off* turns the unused layer off completely. In *Mono* mode, only one note - the last note played - is produced. *MonoLow* generates the lowest note played in the layer's *SPLIT* range. *MonoHigh* also generates a single note, that being the highest note played. In *Paraphonic* mode, all polyphonic voices within the Layer share a single monophonic amplifier envelope. In *Polyphonic* mode, each polyphonic voice has its own amplifier envelope.

Sound is generated by the divide-down oscillators. Each layer has three oscillators tuned an octave apart. Layer 1 has *OSC 16"*, *OSC 8"*, *OSC 4"*. Layer 2 is tuned an octave higher, having *OSC 8"*, *OSC 4"*, *OSC 2"*. The relative volume levels of each oscillator can be controlled to produce a huge array of different timbres. Oscillators can be turned off entirely by setting them to the absolute minimum volume.

The *AMP* envelope consists of a simple attack / release curve that controls the volume of the sound. The *ATTACK* parameter governs how long it takes for the sound to fade in when a note is played. The *RELEASE* Parameter controls how long it takes for the sound to fade out after a note is released.

By turning on *RETRIGGER*, the curve will always start again from the beginning of the attack stage, giving a swelling feel to the sound.

The *VOLUME* control is used to give a boost or cut to the loudness level on the entire layer. This control allows for easier relative volume balancing of Layers 1 and 2. At 0.0dB, no boost or cut is applied to the Layer.

The *VELOCITY* parameter specifies how much the overall volume of the amplifier is controlled by playing dynamics. At 0%, notes will always play at the same volume, At 100%, the amplifier volume is entirely controlled by the MIDI note velocity and the full loudness range, from near silence to very loud, is possible by playing articulations.

Each layer has a monophonic filter section for further sound shaping. The *TYPE* parameter selects the type of filter.

*OFF* bypasses the filter altogether, allowing the raw oscillator sound through.

*LOWPASS*, *BANDPASS*, and *HIGHPASS* are analogue resonant filters with a variety of modulation options. *LOWPASS* allows only frequencies below the cutoff through. *BANDPASS* allows only frequencies around the cutoff through. *HIGHPASS* allows only frequencies above the cutoff through. The *CUTOFF* frequency specifies the point at which frequencies begin to attenuate. *RESONANCE* adds emphasis at the cutoff point. *LFO MOD* controls the amount of cutoff modulation from the *MODULATION LFO*. Likewise, *ENV MOD* control the amount of cutoff modulation from the *MODULATION ENVELOPE*. *AFTERTOUCH* allows monophonic channel aftertouch to modulate the cutoff frequency. Velocity controls the amount that playing dynamics affect the cutoff frequency.

*HP / LP PAIR* are highpass and lowpass filters in series. The *HIGHPASS* parameter con-

trols the point at which low frequencies begin to attenuate. *LOWPASS* controls the point at which high frequencies attenuate. Together, these act as a very flexible bandpass filter.

*9 Band EQ* is a graphic equalizer type filter, with 9 bands at 50Hz, 100Hz, 200Hz, 400Hz, 800Hz, 1.6KHz, 3.2KHz, 6.4KHz, and 12.8KHz. Each frequency band can be amplified for up to 18dB of gain, or -18dB of attenuation.

The *ROUTING* section lets you control the destination of a layer's output. *TO FILTER 2* allows layer 1's post-filter output to be routed into layer 2's filter section. *TO EFFECTS* allow the post-filter output to be routed to the effects section. Note that when *TO FILTER 2* routing is on, layer 1's *TO EFFECTS* button is bypassed.

## SPLITS

The *SPLIT* parameters controls which MIDI notes are sent to which layer. All notes, from the bottom of the MIDI note range up to and including the layer 1 *HIGH SPLIT* point are sent to layer 1. All notes, from the top of the MIDI note range down to and including the layer 2 *LOW SPLIT* are sent to layer 2.

Generally, one of the following setups will be used: - To split the keyboard in half so that all notes up to a certain point go to layer 1, and the rest to layer 2, set layer 1's *HIGH SPLIT* to its crossover point (eg, B3), and layer 2's *LOW SPLIT* to the note above this (eg, C4). - To produce an overlapping stack so that all notes go to both layers, set layer 1's *HIGH SPLIT* to its maximum of G9, and layer 2's *LOW SPLIT* to its minimum of C-1. - You may wish to divide the layers into three so all notes up to a certain point go to layer 1, the next set of notes go to both layers, and the remaining notes go to layer 2 only. To do this, set Layer 1's *HIGH SPLIT* to the layer 1 crossover point (eg, B3), and layer 2's *LOW SPLIT* to the layer 2 crossover point below (eg, C3). In this case, all notes between C3 and B3 will go to both layers.

## MODULATION

The modulation *LFO* can be used to modulate the overall oscillator pitch via the *VIBRATO* section, or the lowpass, bandpass, or highpass cutoff via the filter *LFO MOD*. *RATE* controls the speed of the LFO. When *SYNC* is enabled, the rate is specified in musical subdivisions and synced to the host tempo. With *SYNC* off, the rate is specified in Hertz.

In the *VIBRATO* section, the *DEPTH* parameter controls the amount that the oscillator pitches are modulated by the LFO. When *WHEEL SCALE* is enabled, the vibrato depth is controlled by the modulation wheel, up to the maximum specified by the depth.

The modulation *ENVELOPE* is a monophonic modulation source that can be used to change the cutoff frequency of the lowpass, bandpass, or highpass filters. The amount of change is specified by the filter's *ENV MOD* parameter. The *ATTACK* time governs the time the envelope takes to reach the initial maximum. It then falls to the steady-state

*SUSTAIN* level over the time specified by the *DECAY*. Finally, when all notes have been released, the envelope decays to zero over the *RELEASE* time. When *LEGATO* is enabled, the envelope will not be retriggered by new notes when there are other notes currently being played. This gives the envelope a continuous flowing feel. By turning on *RETRIGGER*, the curve will always start again from the beginning of the *ATTACK* stage when a note is played.

## EFFECTS

The *CHORUS* effect produces the thick ensemble sound of a string synth. String's chorus section is modeled on the bucket-brigade device that characterized the sound of the classic ensemble synths. Select from two different chorus algorithms with the *TYPE* parameter for either an authentic ensemble timbre (*VINTAGE*), or a more modern, denser sound (*MODERN*). The speed of the chorus's pitch modulation is controlled with the *RATE* parameter. The amount of pitch modulation is controlled with the *DEPTH*. Turning this to 0% will bypass the chorus. The *WIDTH* parameter controls the amount of chorus stereo spread. The chorus effect can also be disabled by turning off the power button.

The *PHASER* effect is a sweeping filter that adds moving peaks and troughs in the signal spectrum. The speed of the movement is controlled with the *RATE* parameter, expressed in Hertz when *SYNC* is off, and musical subdivisions when it is on. The *COLOUR* parameter alters the width of the peaks and troughs, as well as the amount of movement, by introducing feedback into the phaser signal. *MIX* is used to balance the original 'dry' signal with the effected 'wet' one. At 0%, only the dry signal is heard; at 100%, only the wet signal.

String's *DELAY* effect produces a series of echoes. String has two independent mono bucket-brigade device delay units, which can be linked to form a single stereo delay. Turning *LINK* on will connect the left and right delay unit controls together so that either set of controls will affect both channels. With *LINK* off, the left and right channels will act independently. The *LEFT TIME* and *RIGHT TIME* parameters govern the time between repeats. This is expressed in milliseconds when *SYNC* is off, and in musical subdivisions when it is on. The amount of signal that is fed back into the delays is controlled with the *LEFT FEEDBACK* and *RIGHT FEEDBACK* parameters. At high levels of feedback, the signal can repeat nearly indefinitely. *MIX* is used to balance the original 'dry' signal with the effected 'wet' one. At 0%, only the dry signal is heard; at 100%, only the wet signal.

## MAIN

The *TUNING* parameter sets the overall tuning. This value is the frequency of the A note above middle C, in Hertz.

The *VOLUME* parameter controls the overall loudness of String's audio output. An output signal that is too quiet can be amplified by increasing the Volume. Likewise, a signal that

is too loud can be attenuated by reducing it.

The VU meters can be used to provide visual cues to the current levels: the louder the signal, the higher the meter will register. Too loud of an output will produce digital clipping, a usually undesirable form of distortion. When this occurs, the meter will display a red warning indicator. Click the meter to reset the warning display.

## String FX Interface

When using the effects only plug-in (StringFX), only the effects section is displayed. The effect parameters function identically to the parameters in the instrument plug-in.



The *GAIN* parameter controls the volume of the incoming signal. Input that is too loud may cause unwanted digital distortion; in the case, turn down the *GAIN*.

## Control Types

### Rotary Controls

Click on a rotary control and drag either upwards to increase the value, or downwards to decrease it. Hold down *SHIFT* and drag to make smaller and more precise changes. Hold down *CMD* (on macOS) or *CTRL* (on Windows or Linux) and click to return the control to its default value. Double-clicking a control will also return it to its default.

Rotary controls can also be changed by hovering the mouse cursor over them and scrolling the mouse-wheel up or down. Holding *SHIFT* whilst scrolling the mouse-wheel will make more finely grained changes.

## Button Controls

Click on a button control to toggle it between on and off. Hold down *CMD* (on macOS) or *CTRL* (on Windows or Linux) and click to return the control to its default value.

## Slider Controls

Click on a slider control and drag either upwards to increase the value, or downwards to decrease it. Hold down *SHIFT* and drag to make smaller and more precise changes. Hold down *CMD* (on macOS) or *CTRL* (on Windows or Linux) and click to return the control to its default value. Double-clicking a control will also return it to its default.

Slider controls can also be changed by hovering the mouse cursor over them and scrolling the mouse-wheel up or down. Holding *SHIFT* whilst scrolling the mouse-wheel will make more finely grained changes.

## Setting the Interface Size

String has a fully resizable interface, allowing it to be scaled to accommodate your own preferences and screen resolution. If supported by the host, String can be resized by dragging the resizer on the bottom-right of the interface. Alternatively, several pre-defined size options can be selected under *options / Zoom*. See also: [configure appearance](#).

The interface size can be restored to the standard size either by choosing *options / Zoom / Default Size* or by holding *SHIFT + CMD/ALT + R* (see [key bindings](#)).

# Preferences

Access *Preferences* by clicking the cog icon (on the top right of the application) and choosing *Preferences...* from the menu.

From here, you can configure the [audio and MIDI devices](#), the [application appearance](#), the [key bindings](#), the default [author name](#), and the channel used for [MIDI learn](#).

## Configure Audio and MIDI

When running String as a standalone application, you will need to configure your audio and MIDI devices. When running as a plug-in, this option is not available because audio and MIDI routing is provided by the hosting application; consult the host's documentation for details. To access the *Devices* configuration panel, select *Preferences...* under the *options* menu (from the cog on the top right of the application).

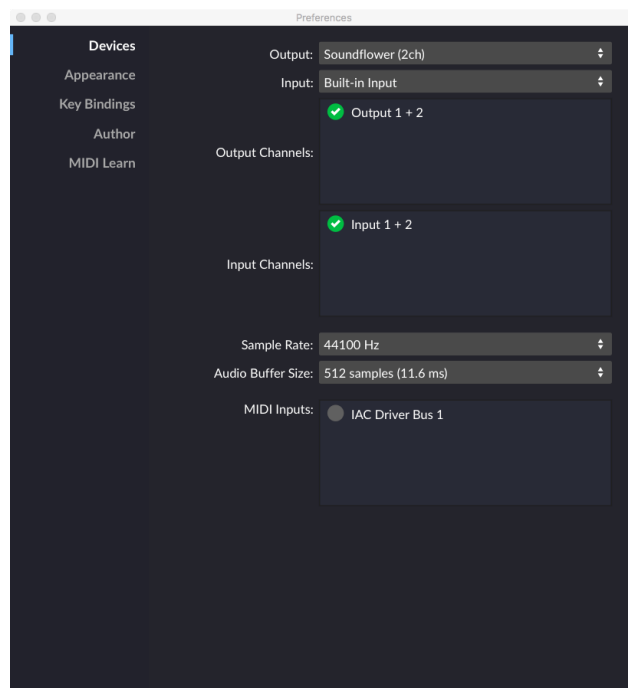


Figure 2: Devices Panel

From this dialog, the following details of your Audio and MIDI setup can be configured:

- Select the *Input* and *Output* audio device(s) you want to use from the list of available devices. Certain devices tend to perform better than others so if multiple devices are available, you should favour: on Mac, using Core Audio; on Windows, using ASIO; on Linux, using JACK.
- For devices that support multiple channels, select the required *Input Channels* and *Output Channels* by checking the box besides the channel name. The String standalone application can process only stereo audio channels.
- The *Sample Rate* dictates the overall quality of the audio output. Using a lower sample rate will reduce the presence of high-frequency components. Higher sample rates should therefore be preferred. However, be aware that the CPU usage is directly proportional to the sample rate. A good compromise between quality and CPU usage is 44100Hz, which is the sample rate of CD audio. Select the sample rate you require from the available list.
- The *Audio Buffer Size* governs latency; a smaller buffer means that String will respond more quickly to MIDI messages and parameter changes, and there will be less delay between input and output. However, a smaller audio buffer size will require more CPU usage. Select the required audio buffer size from the available list. Using too small of an audio buffer can overload your computer to the extent that audible clicks are heard. In this case, increase the audio buffer size until clicks are no longer present.
- Select in the *MIDI Inputs* list any external MIDI devices with which you want String to listen. MIDI devices can be mapped to parameters, and String can also respond to MIDI program change messages.

## Configure Appearance

Changes that affect the appearance of the String user interface are configured here. To access these settings, select *Preferences...* under the *options* menu (from the cog on the top right of the application), and select *Appearance* from the items on the left.

Setting the *Magnification* scales the entire user interface. Any changes made to this option will not take effect until the dialog is closed.

The *Graphics Driver* options allow you to select a different rendering driver. An efficient graphics driver means that less of your processor's CPU is used in rendering the interface, leaving more CPU free to render audio and run plug-ins. Generally, 3d accelerated drivers such as OpenGL are the most performant. On macOS, the CoreGraphics renderer also delivers good performance.

## Theme

Set the colour scheme used for the main user interface. Choose between a *light* and a *dark* theme.

## Configure Key Bindings

Shortcut keys allow often-used actions to be performed quickly from the keyboard using a specific combination of key presses. These actions, and their required key combinations, are listed here. To access these settings, select *Preferences...* under the *options* menu (from the cog on the top right of the application), and select *Key Bindings* from the items on the left.

Each action can have any number of shortcuts assigned to it, including none.

To add a new shortcut key combination to a specific action:

1. Click the + button beside the action to which you wish to add a shortcut.
2. In the *New Key Binding* dialog, hold the key or combination of keys that you want to assign to the action. If these keys are already bound to another shortcut, a warning will appear; despite the warning, the binding can still be made, which may result in ambiguity when the keys are pressed.
3. Press *OK* to make the binding, or *Cancel* to discard the change.

Bindings can be deleted by clicking the *X* button on the key combination that you wish to remove. Some shortcuts are hard coded into the system (such as copy and paste), and cannot be removed.

You can restore all shortcuts to their default values by pressing *Set All To Default*, or restore only some shortcuts by selecting their actions and clicking *Set Selected to Default*.

Some shortcuts may not be available in certain hosts because the hosts themselves consume the key presses. In these cases, redefine the shortcut to something that does not conflict with keys used by the host.

## Configure Default Preset Author

The name entered here will be used as the author name for any newly created presets. To access this setting, select *Preferences...* under the *options* menu (from the cog on the top right of the application), and select *Author* from the items on the left.

When saving a preset, you can override this, but setting it here saves you the hassle of having to put your name into every new preset.



## Configure MIDI Learn

When using an external MIDI device to control String, String can be set to listen to MIDI messages on all channels (*Omni mode*), or on a single channel. To access these settings, select *Preferences...* under the *options* menu (from the cog on the top right of the application), and select *MIDI Learn* from the items on the left.

# MIDI Control

## MIDI Controllers

Instead of configuring a preset using a mouse, you can change any parameter value using a MIDI keyboard or control surface. This is done by mapping each parameter to a MIDI continuous controller (CC). Each parameter can only be mapped to a single MIDI controller; the same MIDI controller can, however, be mapped to several different parameters.

MIDI controller mappings are global; once defined, the same mapping assignments will be used by all String Programs. This means that you only need define mappings once for your particular controller, and it will be usable in all your projects.

## Display Current MIDI Controller Mappings

To view the current MIDI controller mappings:

1. Enter *MIDI Learn Mode* by choosing *Start MIDI Learn* under the options menu. Each parameter's mapped MIDI controller will now be displayed on an overlay. If room allows, the channel will also be displayed, or X for omni-channel mappings.
2. When you have finished viewing the MIDI controller mappings, choose *Stop MIDI Learn* under the options menu.

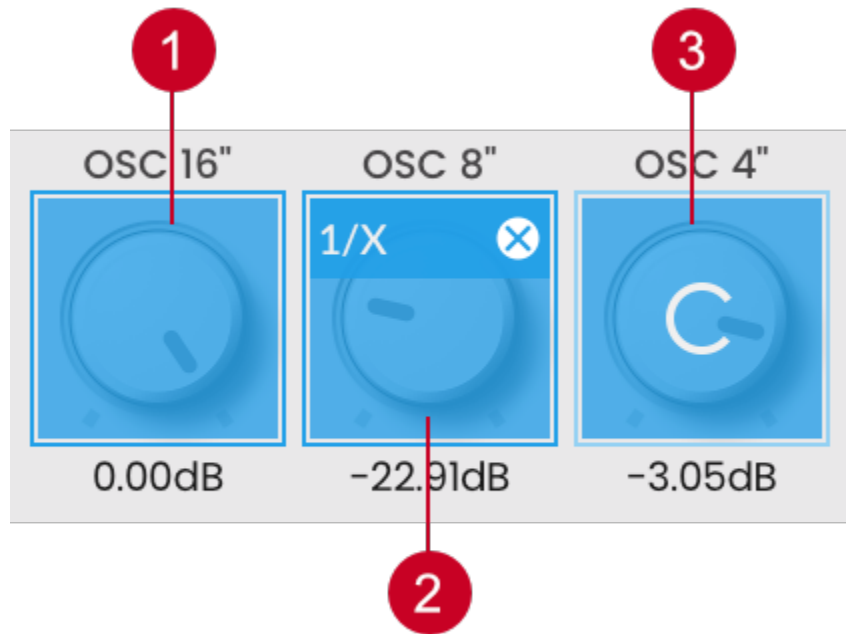


Figure 3: MIDI Mapping. (1) Unmapped Control; (2) Mapped Control; (3) Control Currently Being Mapped

## Assign MIDI Controllers to Controls

To map a MIDI controller to a parameter:

1. Enter *MIDI Learn Mode* by choosing *Start MIDI Learn* under the options menu.
2. Click on the parameter you wish to map to a MIDI controller.
3. Turn your MIDI controller. The parameter will now be labelled with the MIDI CC number of this controller.

You can now either:

- Map another parameter by repeating these steps from Step 2.
- Finish mapping parameters by choosing *Stop MIDI Learn* under the options menu.

## Remove MIDI Controller Mappings From Controls

To remove a mapped MIDI controller from a parameter:

1. Enter *MIDI Learn Mode* by choosing *Start MIDI Learn* under the options menu.
2. Click on the close button of the parameter from which you wish to remove the mapping. This parameter will become unmapped, and the CC number label on the pa-

parameter will disappear to reflect this.

You can now either:

- Remove the mapping from another parameter by repeating these steps from Step 2.
- Finish removing parameter mappings by choosing *Stop MIDI Learn* under the options menu.

## MIDI Program Change Lists

You can change presets by using *MIDI program change* commands. This requires a MIDI keyboard or a control surface with the ability to send such MIDI messages; consult your controller's documentation for details. If using String as a plug-in, this functionality is dependent upon the host correctly forwarding MIDI messages to the plug-in; check the documentation provided with your host for details.

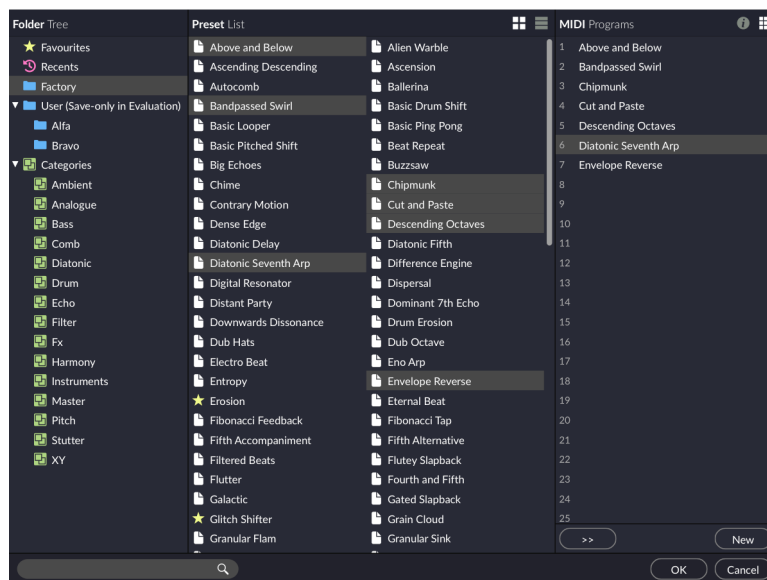


Figure 4: MIDI Program Change List

To select which presets are loaded in response to MIDI program change commands, you will need to create a MIDI program change list. Select *Edit MIDI Program Change List...* from under the options menu to display the MIDI program change list editor.

To add presets to the MIDI program change list, either double-click on a preset, or select a Preset and click the » button. Use the *info* and *list* button on the top-right of the window to toggle between viewing the list, and information on the selected preset.

Any presets in the MIDI program change Lists are automatically loaded and cached in memory. This means that Preset changes can be instant. There may be a slight pause after editing the MIDI program change list whilst the presets within it are loaded. When a

preset is saved, any entries in the MIDI Program Change List are saved alongside it. Note that setting a preset with a MIDI Program Change List as the [default preset](#) will cause all presets within the list to be loaded when a new instance of String is loaded.

# Preset Management

## Parameters and Presets

You can modify the type of sound that String produces by setting the values of parameters. Each parameter will affect a specific element of the sound. The state of all parameters is collectively called a preset. Presets are given a suitable name, and the name of the currently selected preset is shown in the menu bar. If the preset has been modified without being saved, a \* appears beside the name.

## Navigation Controls

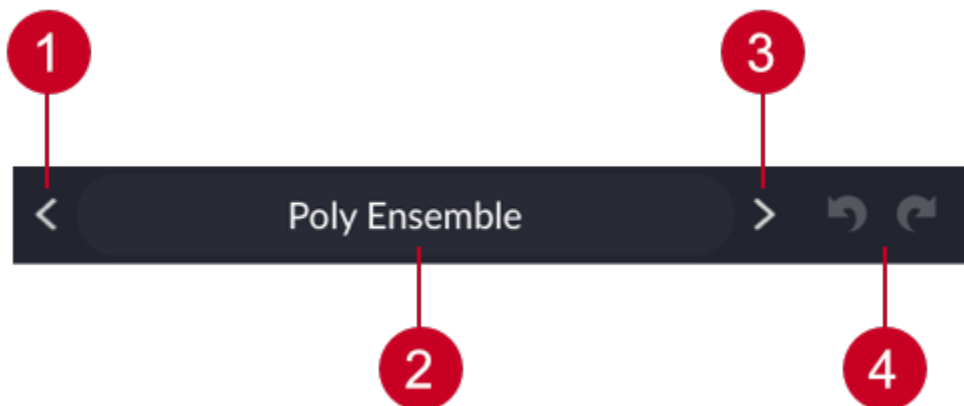


Figure 5: Navigation Controls. (1) Previous Button; (2) Preset Name; (3) Next Button; (4) Undo/Redo Buttons

If you wish to browse quickly through the presets, you can do so using the the *previous* and *next* controls. These controls are context sensitive, only displaying presets that are relevant to the last [search](#) you made. For example, if you had most recently searched for “analogue, bass”, only presets that match this will be selected. Scrolling the mouse-wheel when the mouse cursor is over the *preset name* will also load the next or previous preset, depending on the scroll direction.

Clicking on the *preset name* opens the [preset browser](#). By pressing *undo*, you can take back the most recent change made to a preset. *Redo* will re-apply the last undo changes. Note that only changes made via the String interface can be undone; changes made externally (such as from a host application, or from a MIDI controller) are not undoable.

## Creating a New Preset

Select *New* to create a new preset. This preset will be in the default initialised state. If you've made a sound worth keeping, be sure to save the preset before you create a new one, as the undo history will be discarded when the preset state is reset.

## Comparing a Changed Preset to the Original Preset

Often you'll want to compare a preset that you have amended to see if it is actually an improvement on the original. Pressing the *AB* button temporarily reverts a changed preset back to its original state. Pressing the *AB* button a second time returns to the changed preset. This method allows you to easily contrast the preset changes you have made to the original preset state.

## Opening Presets

To select a preset from within your preset library, click on the preset name in the menu bar to open the preset browser.

## Saving Presets

Click *Save* to save the current preset. Choose the location in which to save the preset in the folders window within the [preset browser](#). Click *OK* to actually save the preset.

## The Preset Browser

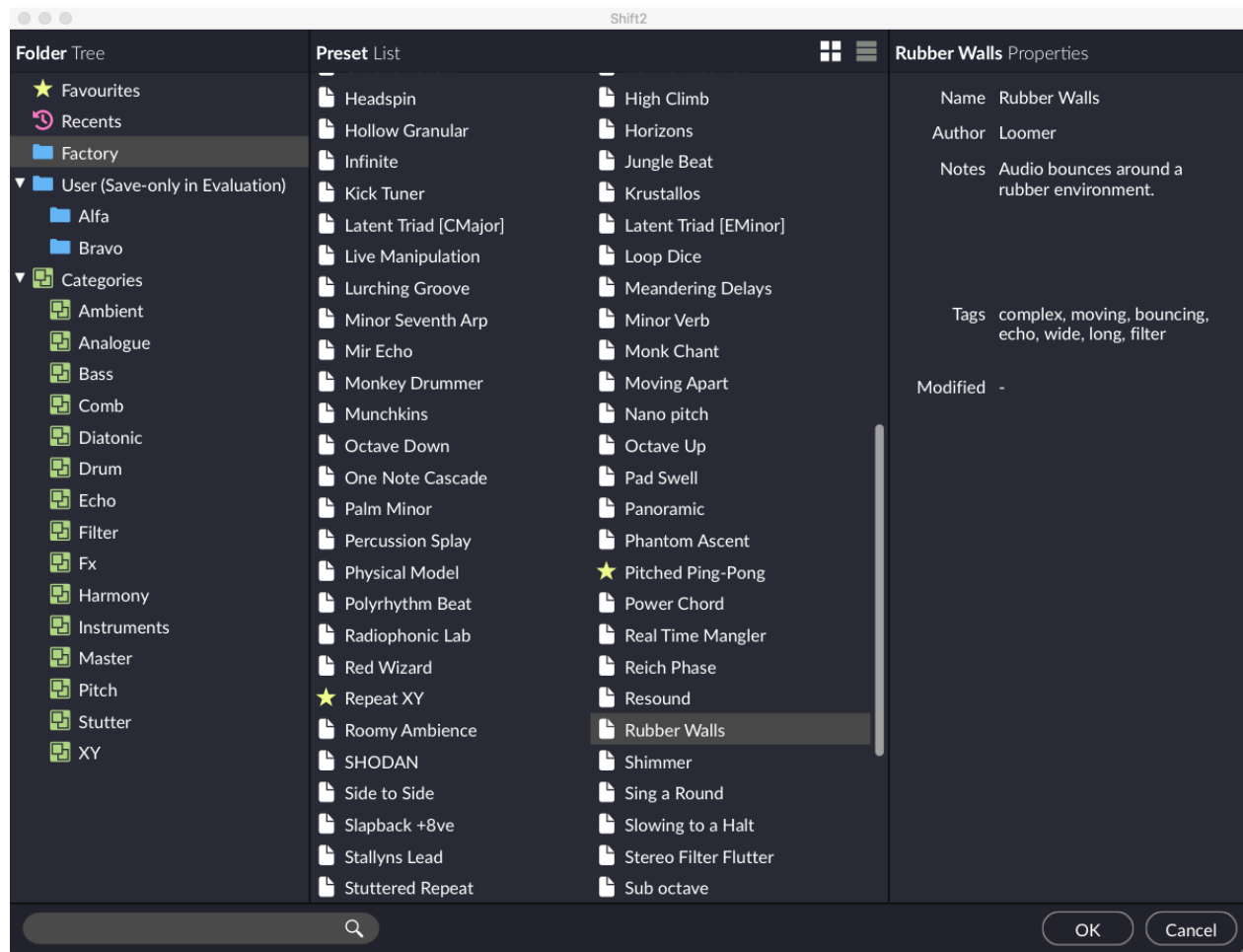


Figure 6: Preset Browser

Presets can be divided into two groups; factory, and user. Factory presets are a collection of example presets built into String. Factory presets can't be overwritten, although you can, of course, edit and save your own presets derived from factory presets.

User presets are stored as individual files on your computer's hard drive. They are found in the following locations:

- **macOS:** `~/Library/Application Support/Loomer/String/Presets`
- **Linux:** `$XDG_DATA_HOME/Loomer/String/Presets`
- **Windows:** `Documents\Loomer\String\Presets`

Only registered users can open user presets. Unregistered versions of String are limited to the factory presets. Unregistered versions of String can, however, still save presets. Any presets created in the unregistered evaluation of String can be opened by purchasing a licence for String.



## Preset Hot-swap

Selecting a preset in the browser will automatically load the preset: this functionality is called *preset hot-swap*. Preset hot-swap allows you to quickly preview presets in the context of the current project. If you decide to use the selected preset, close the browser by pressing *Return* or by clicking *OK*. To return to the preset you had open before you accessed the browser, close the browser by pressing *Escape* or by clicking *Cancel*.

## Preset Meta Data

Presets can have meta data attached to them. Meta data doesn't influence the sound of the preset, but is used in categorizing, searching, or annotating them. The following meta data is available:

- **Author:** The name of the preset creator. You can enter your name into the [preferences dialog](#), and that name will automatically be used as the default author for all new presets.
- **Notes:** Any additional notes about the preset, such as influences, playing tips, description, etc.
- **Tags:** Tags are short words that describe the preset. Examples are: "bass", "distorted", "monophonic", "trance", "quiet", etc. Tags are used to help quickly search for a specific sound.

## Searching for Presets

Type keywords into the search field to locate all presets that match this description. The search will match the keywords in the Preset Name, Author, and Tags fields, and parent folder name.

## Categories

Categories allow you to quickly access a subset of your presets. Categories are listed under the categories node in the preset browser. A category is simply a shortcut to searching for the category name; a category called "Bass" will list all presets that match the search criteria "Bass". You are not limited to create categories based on the type of sound: you can, for example, create categories for preset authors, or indeed for any other search term. Provided you tag your own presets, they will automatically be added to the correct category (or categories: presets can belong to multiple categories; a preset could be both a "Bass" and "Lead" sound, for example.)

## Preset Formats

Presets in the user presets folder need to be in either native .xml format, or VST .fxp or .fxb format. Standalone, and Audio Unit versions can also open .fxp or .fxb files within the browser. Please ensure any files in the user presets folder have the correct file extension.

As a general rule, you should prefer saving presets in String's native .xml format over any proprietary host format, as this can make it easier to share presets with users on other hosts.

## Copying and Moving Presets

Presets can be moved or copied to another folder by dragging and dropping a preset onto the destination folder. By default, presets will be moved. You can instead copy the preset by holding shift whilst dragging. Note that presets inside banks (either .xml or .fxb banks), can only be copied, not moved. Multiple presets can be moved or copied at once by selecting more than one preset.

## The Default Preset

The default preset is automatically opened when a new instance of String is created. To specify the default preset, right-click on a preset and choose *Set as Default*. Note the AAX plug-ins do not support default presets: a new instance of String as an AAX plug-in will always begin with an initial preset state equivalent to a new preset.

## Restoring Deleted Factory Presets

To reinstate any deleted factory presets, right-click on the factory branch in the [preset browser](#) and choose *Restore Factory Presets*.

## User Folder Management

Folder management is performed by right-clicking on a user folder, and selecting an option from the pop-up menu. From here, you can create a *New Folder*, *Rename Folder*, or *Delete Folder*. On Windows or macOS, you can also quickly locate this folder in Finder or Explorer by choosing *Show In Finder / Explorer*.

## Favourites

Favourite presets can be quickly located under the *Favourites* node in the preset browser. A preset can be added to or removed from the favourites list by right-clicking and choosing *Add to Favourites* or *Remove from Favourites*, respectively.

# Status display

Status display information is displayed at the bottom right area of the String interface:

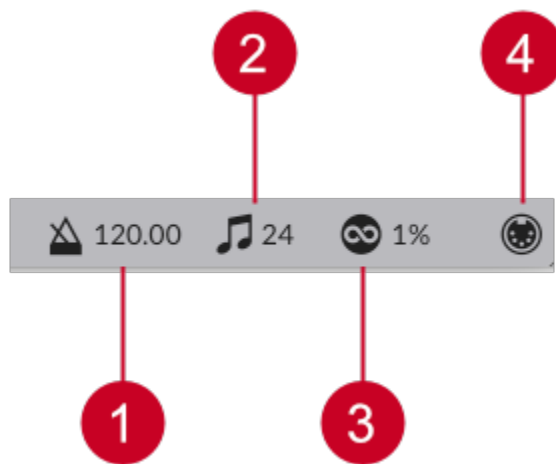


Figure 7: Status Display. (1) Tempo; (2) Polyphony; (3) CPU Usage; (4) MIDI Input

## Tempo

The *tempo control* only appears when running String as an application; when running as a plug-in, the host application will provide the tempo for String. The tempo control displays the current tempo, in beats per minute (bpm). It can be changed by either:

- Double-clicking the tempo control, typing the required bpm and pressing *Return*.
- Clicking on the tempo control and dragging either upwards to increase the tempo, or downwards to decrease the tempo.

## Polyphony

This shows the number of currently playing voices.

## CPU Usage Monitor

The *CPU usage monitor* displays the percentage of processor resources dedicated to generating the audio output. The total processor resources on a computer are limited, so when String increases its CPU usage, there will be less CPU resources remaining for other plug-ins and applications. String's CPU requirements will differ depending on which preset is currently loaded and how that particular preset is configured.

A value of 100% indicates that your computer is running at maximum capacity and will not have enough processing power remaining for anything else. You might notice audio output become temporarily glitchy, or that the user interface becomes sluggish. There are several methods for resolving this:

- Choose a different preset that has lower CPU usage.
- Reduce the number of voices in the current preset.
- If running in standalone mode, you can increase the latency or decrease the sample rate. See section [Configure Audio and MIDI](#).
- If running as a plug-in, changing your host's audio device settings can improve performance.
- If your host offers freeze or offline processing, you might be able to render the audio output in non-realtime.

## MIDI Input Monitor

The *MIDI input monitor* flickers to show that a MIDI message has been received. You can use this to confirm that your MIDI controller is configured correctly for String.

# Contact Details

## Online

**Website:** [www.loomer.co.uk](http://www.loomer.co.uk)

**Support and FAQ:** [www.loomer.co.uk/support.htm](http://www.loomer.co.uk/support.htm)

## Email

**Sales information:** [sales@loomer.co.uk](mailto:sales@loomer.co.uk)

**Technical support and feature requests:** [support@loomer.co.uk](mailto:support@loomer.co.uk)