

# Unify 2 User Guide



Document version 1.02

# Table of contents

<b>Welcome</b>	<b>7</b>
<b>Unify 2</b>	<b>8</b>
<b>Getting started with Unify: Overview</b>	<b>9</b>
What is Unify?	9
Unify is a virtual instrument	9
Unify is both a plug-in AND a plug-in host	9
Unify includes its own suite of plug-ins	10
Unify is a platform for new sound libraries	10
Unify can make your third-party plug-ins easier to use	11
Layers: How Unify creates composite sounds	12
Signal Flow	14
MIDI Manipulation on MIDI and Instrument Layers	15
<b>Getting started / Installation</b>	<b>16</b>
Download the Installer and Unify 2 Standard Library	17
Installation	18
Activate and set up Unify 2	19
Run the standalone app	19
Note for existing Unify 1.x users	19
Activate your license	20
Online authorization (recommended)	21
Offline authorization	22
Activation completed	23
Install Factory Content	24
Final Set-up steps	27
What to do if you have problems	29
<b>Unify 1 to Unify 2 upgrade - important notes</b>	<b>30</b>
Installation	30
Compatibility and shared library usage	30
Licenses auto copy - Unify 1 to Unify 2	30
<b>UI Overview &amp; Navigation</b>	<b>31</b>
Sections overview / Quick start	31
Layers View - Intro	31
Play mode - Intro	32
Settings View - Intro	33
Top bar / meta-data view	34
Top - 'Burger' menu:	34
Top bar - Settings button	34

Top bar - Add layer button	34
Top bar - Meta-data area	34
Top bar - Patch navigation, master volume and CPU meter	35
Settings view - details	36
UI Settings	36
GUI	36
GUI style options	37
Standalone app options:	38
Plugin windows:	38
Parameter links:	38
Embedded Unify:	38
On patch load / init:	39
Transport:	39
Audio / MIDI:	40
Paths / Data	42
Plugins	44
Scanning your system for plug-ins	45
You can also have Unify scan your system for available plug-ins.	45
Note that when scanning for plug-ins, Unify actually has to try to load each one.	45
This will trigger any copy-protection mechanisms a plugin uses.	45
Registering individual plug-ins	48
Plug-in scanning issues and work-arounds:	49
Sub-sets	50
Using the sub-set editor	51
Licenses	52
Layers view - details	53
Layers signal flow	54
Layer controls	55
Layer plugin boxes	56
MIDI and Instrument layer input controls	58
MIDI cc filters and the Latch button	59
MIDI and Instrument layers - zones editor/live keyboard view	61
Layer audio controls	62
MIDI out	63
Layer drag-and-drop	64
Patch browser	65
Side by side mode	66
Searching	67
Patch loading modes and right click options	68
Transport & controls panel	70
Virtual keyboard / MIDI controls	73

Macro controls panel	75
Play mode view	76
Recording controls panel	82
Metronome options panel and the metronome	83
<b>Creating / Editing new sounds</b>	<b>84</b>
<b>Creating your first patch (Tutorial)</b>	<b>84</b>
Start from an INIT patch	84
Create the piano layer	85
Add the synth layer	89
Adjusting the layer balance and octaves	91
Save your new patch	92
<b>Macro controls, modulation and parameter links</b>	<b>94</b>
<b>Controlling Unify using DAW automation</b>	<b>95</b>
<b>Advanced usage of the Metronome Unify</b>	<b>97</b>
<b>Layer audio input and output / layer MIDI out</b>	<b>100</b>
Selecting an audio input	100
Selecting an audio output	101
MIDI out	102
Hardware synth	102
<b>The built in and bundled plugins</b>	<b>103</b>
Instruments	103
MIDI Effects	104
Audio Effects	104
Delay	104
Distortion	104
Dynamics control	105
Equalization and Filtering	105
Modulation	105
Reverberation and Ambience	106
Special	106
<b>Guru Sampler 2</b>	<b>106</b>
Sample panel / Selecting samples	108
Importing Samples	108
Filter controls	112
Envelopes / envelope curves	112
Assigning modulations	114
Patch conversion from Gurusampler 1	116
<b>NoteMate</b>	<b>116</b>
Using NoteMate in a Unify2 patch:	119
<b>Step Sequencer</b>	<b>120</b>
Step editing:	121

Per step buttons:	122
Keeping the sequence running:	122
Timing options:	122
<b>Appendix / Advanced considerations</b>	<b>123</b>
Keyboard shortcuts	123
General	123
INIT button	124
Selecting patches in the patch browser	124
Ops (operations) buttons	125
Plug-in ops buttons	125
Layer ops buttons	125
Curve editors	125
Number boxes	126
Parameter sliders/knobs	126
Layer level sliders	126
Macro parameter knobs	126
Parameter knobs in built-in plug-ins	126
Virtual Keyboard view	126
Multi-selection lists	127
OSC support	128
Overview	128
Enabling and Configuring OSC	128
OSC message types	130
Parameter-set messages	130
Command messages	130
Bidirectional messaging for patch-loading commands	131
MIDI-over-OSC messages	132
OSC messages to embedded Unify instances	132
Key Zones	133
Defining key-zones using a text file	134
Selecting a key-zone file when saving a patch	136
Parameter paths reference	137
Basic concepts	137
Paths, Components, and Values	137
Path validity	137
Path Categories	138
Top-level paths	138
Macro paths	138
Layer-specific paths	139
Plug-in parameters	140
Patch Variations	141

Basic Concept	141
Example 1: Song sections	142
Variations file	143
Parameter links	143
Example 2: Sample maps	144
Variations file	144
Sample-names list files	145
Parameter links	148
Important reminders	149
About .guru files	150
Example .guru file	150
How Unify unzips the .guru file	151
Limitations of .guru files	151
Creating a .guru file (the safe way)	151
Creating a .guru file (the risky way)	152
Creating a .guru file (more generic)	152
Testing your .guru files	152
Keeping Unify's patch database clean with scrub.xml	154
Location and format of the scrub.xml file	154
Structure of the scrub.xml file and rules	155
Unify's files and folders	156
Four distinct groups of files	156
Finding Unify's files on a Windows PC	156
Finding Unify's files on a Macintosh	158
Troubleshooting basic file/folder issues	159
Leftover .guru files	159
<b>Unify 2 vs Unify 1 - what's new?</b>	<b>160</b>
Layers:	160
Play mode:	161
Browser:	161
Bottom bar/transport area/recording/metronome:	162
The metronome:	162
Recording:	162
New features for the macro controls:	163
New plugins (GuruSampler2 +)	163
Gurusampler 2:	163
New audio analyzer plugin	165
New MIDI step sequencer plugin	165
Updated audio file player plugin	166
Plugin window features:	166



# Welcome

Thank you for purchasing Unify 2.

Please make sure to read the readme file and quickstart guide before starting the process of installation.

More detail about installation is provided in this document.

Note - unless describing something specific about Unify 2 or Unify 2 vs Unify 1 we may refer to Unify 2 as simply 'Unify' in some sections of the document.

To learn about what's new in Unify 2 vs Unify 1 please refer to the relevant section in the Appendix.

# Unify 2

If you own Unify 1, then you probably already have a pretty good idea of what makes Unify so powerful and unique. If you're new to Unify 2, welcome, we'll start with some information about what makes Unify unique in the world of plug-ins.

Unify 2 marks the next evolution of PlugInGuru's acclaimed Unify (Standard Edition), originally conceived by John "Skippy" Lehmkuhl and developed by Shane Dunne with Skippy working with Unify throughout development to make sure it worked the way he imagined it should work.

In a rare and serendipitous turn of events, Shane introduced John to Paul Carter — the developer behind the celebrated Hyperion modular synth, Theia, and numerous sound libraries under his Wavesequencer brand. The collaboration was a natural fit, given John's own history as a sound designer and contributor to numerous Korg synthesizers including the legendary Wavestation and its pioneering wavesequencing technology.

Since January 2025, Paul and John have been working closely to reimagine and expand Unify, bringing a host of new creative features, enhanced performance, and an even more intuitive user experience to Unify 2.

At its most basic core, Unify (V1 and V2) is a plug-in host that can load all of your Instruments, MIDI Effects and Audio Effect plug-ins and work with them in a simple, intuitive way. It works with a graphical user interface centering around layers. Where Unify goes off the rails is that it is also a sonic playground on its own BEFORE you add your plug-ins into the Unify universe. We have received permission from numerous companies to include their wonderful free public domain based Instruments, MIDI Effects and Audio Effects so that out of the box Unify includes more than 50 plug-ins used to make over 1,100 Patches in the factory library.

With Unify 2 we have added a number of new features, as have implemented a complete UI re-design, aimed to make things more straightforward, visually intuitive and attractive.

Unify 2 has new built-in plugins and features, including a more advanced sample player, a step sequencer, recording, audio in/out and MIDI output capabilities.

The 32 Macro Knobs from Unify V1 have been improved with a modulation engine more powerful than an LFO and there is even a new metronome feature that can play an entire multi-layer Unify instance as your backing groove/drum track (or whatever else you can think of!)

Unify 2 also features a new 'play mode' for hiding the layers view so you can focus on just playing. To create an inspiring compositional workspace, built-in background images can be assigned to presets, while large macro control knobs provide quick access to the major sound shaping controls you can set up for each patch.

# Getting started with Unify: Overview

## What is Unify?

### Unify is a virtual instrument

Unify 2 is a *virtual instrument*, which comes with over 1300 presets, and over 8 GB of samples (which are a small taste from over 24 Unify V1 libraries.) An ever-expanding series of add-on preset libraries is available from [PlugInGuru.com](http://PlugInGuru.com) and others.

Unlike most virtual instruments, Unify creates sounds by “patching” multiple other plug-ins together in a semi-modular environment. It comes with a collection of built-in and bundled plug-ins, which includes instruments, audio-effects, and MIDI-effects. Unify can also load just about all the plug-ins you already own (VST and VST3 on Windows, plus Audio Units on Mac), and every combination sound you create can be saved as a preset or “patch” for quick recall later.

Think of Unify as a super-powerful Combination mode as in flagship music workstations, where you can use ANY instruments and effects, not just the ones provided by a particular company. Unify has an unlimited number of Layers which can contain an Instrument and zero or more audio effects, and can optionally be controlled by one of an unlimited number of MIDI effects.

There are up to 4 AUX Busses and all audio is finally routed to a Master Effect Bus. (See “signal flow” below.)

### Unify is both a plug-in AND a plug-in host

Unify is a *stand-alone app*, as well as a plugin ( VST3, and Mac Audio Unit) you can use in a DAW app like *Logic Pro*, *Cubase*, *Ableton*, *Bitwig* etc.

The app and plug-in versions are mostly identical; the app version gives you the convenience of playing without a DAW, e.g., for live performance and provides direct access to audio hardware IO rather than through virtual inputs and outputs.

Like a DAW, Unify can *host* multiple plug-in *instances*, and connect them in various ways to produce composite sounds. It allows you to easily connect plug-ins in ways that are much more difficult (or in some cases, impossible) in a DAW.

For example, most DAWs provide very limited support for MIDI-effect plug-ins (which don't process audio at all, only streams of MIDI data), but Unify's MIDI effect support is excellent and very flexible.

Also like a DAW, Unify can draw on all the processing power in modern *multi-core* CPUs.

This doesn't magically make your computer more powerful. Multiple *layers* in Unify are pretty much the same as multiple *tracks* in your DAW, but it does allow you to define, save, and quickly recall multi-layer combinations without losing the multi-threading benefits which your DAW provides.

You will be able to load instances of just about all of the VST2/VST3 and AU plug-ins you own in Unify, just as you can in your DAW. In some cases, this will allow you to use plug-ins which your DAW *cannot* load directly, e.g., *Logic Pro X* can only load Audio-Unit plug-ins, but you can use the AU version of Unify to load VST2/VST3's and use them in *Logic*.

Unify is ideal for using *MIDI-effect plug-ins*, which can be clumsy in many DAWs, and impossible in some.

## Unify includes its own suite of plug-ins

You can use your own purchased plug-ins with Unify, but you don't have to, because Unify comes with a whole suite of plug-ins. All the factory patches in the Unify Standard Library are built using *only* these included plug-ins, so you can get started playing them right away.

Unify's plug-in suite is divided into two groups:

- **Built-in plug-ins** are part of Unify itself. These include Guru Sampler and Guru Sampler 2, a suite of *audio effects*, and a collection of powerful and unique *MIDI effects*.
- **Bundled third-party VST plug-ins** are plug-ins created by other developers, who have kindly given their permission for them to be included with Unify.

## Unify is a platform for new sound libraries

Unify is also the vehicle for most new *PlugInGuru.com* patch libraries. Thanks to Unify's simple but effective copy protection, libraries for Unify are not posted online by software pirates on ebay for pennies.

Since Unify 1 launched in January of 2020, we have shipped over 40 paid libraries and, at the time of this writing, 7 free libraries. We have aggressively added new features and abilities to Unify that the new libraries take advantage of. A number of other sound designers have also begun to offer patch libraries for Unify. Many of these are "Unified" versions (Unified meaning a patch saved in Unify format that is a patch from a library for another Plug-In) of their existing library for a totally different plug-in because of the power Unify Patches have.

## Unify can make your third-party plug-ins easier to use

The largest class of free libraries for Unify are “unified” libraries. A “unified” library for a specific third-party virtual instrument plug-in has one Unify patch for every one of that plug-in's factory presets. You can quickly load an instance of that plug-in, with a specific preset ready to go. In most cases, unified patches also include effects and pre-assigned macro knobs, and most unified libraries include “bonus patches” where one or more instances of the “unified” plug-in used together with Unify's built-in and bundled plug-ins, to create huge combination sounds.

Unify has several unique abilities which can make loading third-party plug-ins inside Unify more compelling than loading them directly in your DAW:

1. Many plug-ins load more quickly inside Unify.
2. Because Unify layers run on separate CPU cores, “combo”/“multi” patches made using multiple instances of a plug-in will often work better than the same combinations created within a single instance.
3. With Unify's built-in PolyBox MIDI-effect, you can increase the polyphony of any virtual instrument, and even play *monophonic* plug-ins polyphonically using multiple instances.
4. Unify's Macro Knobs let you control any number of plug-in parameters in real-time, and map the most important ones (or combinations) to your own MIDI controller(s) in consistent ways.
5. Because Unify can *load instances of itself*, creating layered and split combinations by loading entire patches (including “unified” factory presets) is quick, easy, and fun.

# Layers: How Unify creates composite sounds

Unify creates composite sounds (called *patches*) using a system of *Layers*, of which there are three distinct kinds:

1. **MIDI layers** host *MIDI-insert plug-ins*
2. **Instrument layers** host a chain of plug-ins as follows:
  - zero or more *MIDI-insert plug-ins* (which modify MIDI data)
  - exactly one *Instrument plug-in* (which produce stereo audio in response to incoming MIDI)
  - and a chain of zero or more *Audio-effect plug-ins* (which process stereo audio)
3. **Audio Effects layers** contain only a chain of zero or more *Audio-effect plug-ins*
  - There is always exactly one *Master Effects layer*, into which the sounds from all other layers are mixed.
  - Some Unify patches may also use *AUX Effects layers* which are comparable to “aux bus” channels in a DAW or a traditional mixing console.

**The simplest Unify patch** would consist of just one Instrument layer. Incoming MIDI is sent to the hosted Instrument plug-in, which produces corresponding notes, and its output might then be processed through one or more audio-effect plug-ins.

**Adding a second Instrument layer** allows playing two instruments at once, each with its own chain of audio effects if you wish. You can continue, adding as many instruments as you wish. This is traditionally called “layering”.

**Key splitting:** Suppose you have two Instrument layers. You can restrict each layer's MIDI note-range so that, for example, layer #1 responds only to the left-hand part of your keyboard, and layer #2 to the right-hand part. This is called a **key split**, or simply **split**. For example, you might want a bass sound in the left hand, and a flute sound in the right hand. You might then consider adding a third piano layer, which is not split, to create a **combination split/layer** arrangement.

**Velocity splitting:** Another option is to split two or more layers based on MIDI note velocity (**velocity split**). You could, for example, set Layer 1 to play a dark, mellow piano sound, and Layer 2 a very bright sound. By setting Layer 1 to respond to MIDI note-velocities below, say, 80, and Layer 2 to respond to velocities higher than that, you can create a velocity split which is more dynamically expressive than either of the two individual sounds.

**Master Effects layer:** Unify's *Master Effects layer* processes the entire composite sound of a patch. This is an ideal place to put effect plug-ins to shape the overall tonal balance (e.g. equalizer) and dynamics (e.g. compressor, limiter) of the sound.

**AUX Effects layers:** Putting audio-insert effects on each individual Instrument layer can be useful, but often you would prefer to have just one or two effect chains which are shared among several Instrument layers. Unify provides *AUX Effects layers* for this. As soon as you create the first AUX

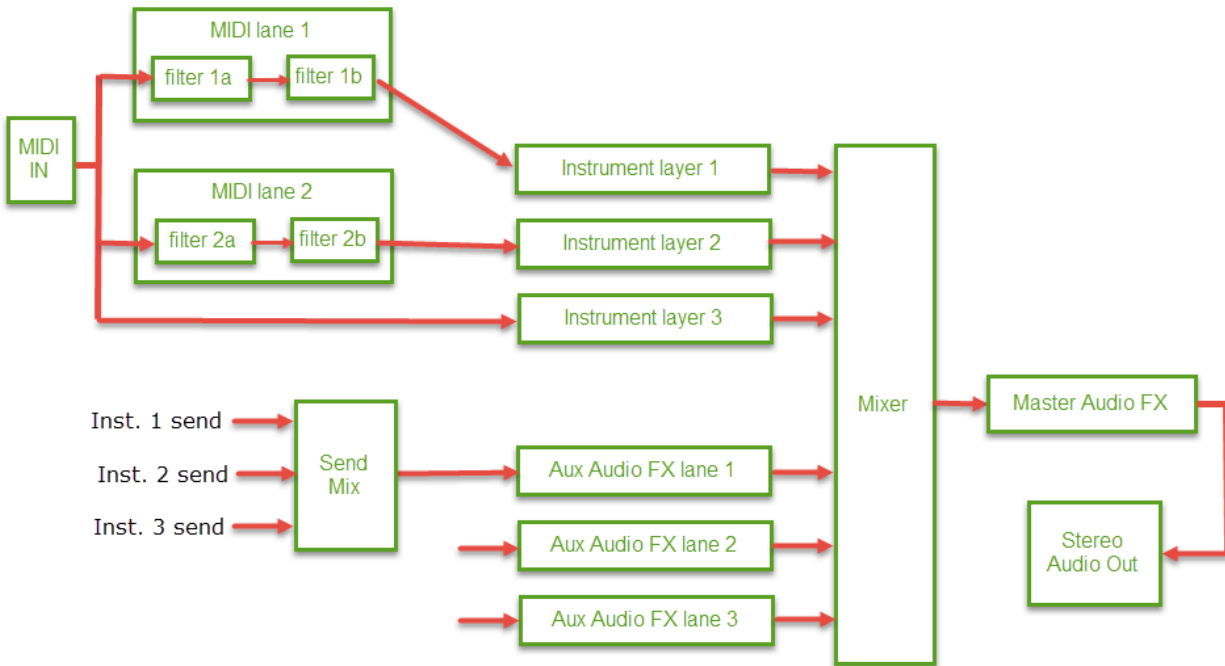
Effects layer, Direct and AUX send controls appear on all Instrument layers, so you can separately control how much of that layer's output goes directly to the Master Effects layer, and how much goes to each AUX Effects layer.

**Mixing:** Every Instrument and AUX Effects layer in Unify has its own **level** and **pan** controls, to control how they are mixed into the Master Effects layer (and hence to the main stereo output).

**Multi-Threading:** Unify automatically allocates each layer to a different core in a multi-core CPU, allowing it to draw on all your computer's processing power. Think of *each layer in Unify* as roughly equivalent to a *track* in your DAW. *One instance of Unify* running, say, ten layers would tax your computer's CPU power about as much as *ten individual tracks* loaded with the same set of plug-ins.

# Signal Flow

This diagram shows the basic signal flow inside Unify:



- Incoming MIDI data (from your DAW and/or MIDI keyboard/controller) is routed to all MIDI layers.
- Each MIDI layer can host a whole chain of MIDI effects, with the MIDI output of the first one being the input to the second, and so on, and the output of the last MIDI effect being the output MIDI stream for that layer.
- Each Instrument layer can be set to receive MIDI directly from the MIDI input stream, OR from the output of any selected MIDI Layer.
- If AUX layers are used, each Instrument layer will show one *Aux Send* knob for each, which controls how much of that Instrument layer's output is sent to each AUX layer, and also a *Direct Send* knob, to control how much is sent directly to the Master Effects layer (bypassing the AUX layers entirely).
- Unify computes a stereo mix of all Instrument and AUX layers, based on each layer's *level* and *pan* controls, and this mix goes to the input of the Master Effects layer.
- The output of the Master Effects layer is the output of Unify itself.

## MIDI Manipulation on MIDI and Instrument Layers

Unify also provides some additional MIDI-manipulation options, which are not shown in the diagram:

- The selected MIDI input stream for each Instrument layer is routed to every plug-in on that layer, including the audio effects, allowing for MIDI control of effects, if desired.
- Each MIDI and Instrument layer can perform certain manipulations of the input MIDI stream, even if no instrument plug-ins are loaded. These include:
  - Filtering based on MIDI channels, and altering the channel-number
  - Transposing MIDI pitch up or down by semitones (up to four octaves)
  - Filtering based on MIDI pitch (for key splits) or velocity (for velocity splits)
  - Altering MIDI note-velocities based on a custom velocity-response curve
  - MIDI Latching: incoming notes can be latched, so the note toggles ON and OFF with successive note-on events. Both monophonic and polyphonic latch modes are available.

# Getting started / Installation

Please follow this guide to install and get started with Unify 2.

Installing Unify and being ready to play sounds requires 5 steps:

1. Downloading the application installer and content .guru files (5 files in total)
2. Run the installer (we recommend using the default install location)
3. Launch the standalone application and unlock with your product key.
4. Sequentially drop the .guru content files into the application.
5. Pick your audio interface settings and MIDI controller device in the audio settings tab so you can hear Unify 2 and play it with your controller(s) of choice.

## Download the Installer and Unify 2 Standard Library

Unify 2 is provided as a zip file for Mac or Windows OS containing the application installer binary and the quick start and readme files.

Also provided are the 4 content .guru files.

Your receipt email contains the links to access these files and also has your License Key to unlock Unify 2 - this information and the files are also available in your account at [PlugInGuru.com](http://PlugInGuru.com)

Download and extract the installer zip and the 4 content files before proceeding.

# Installation

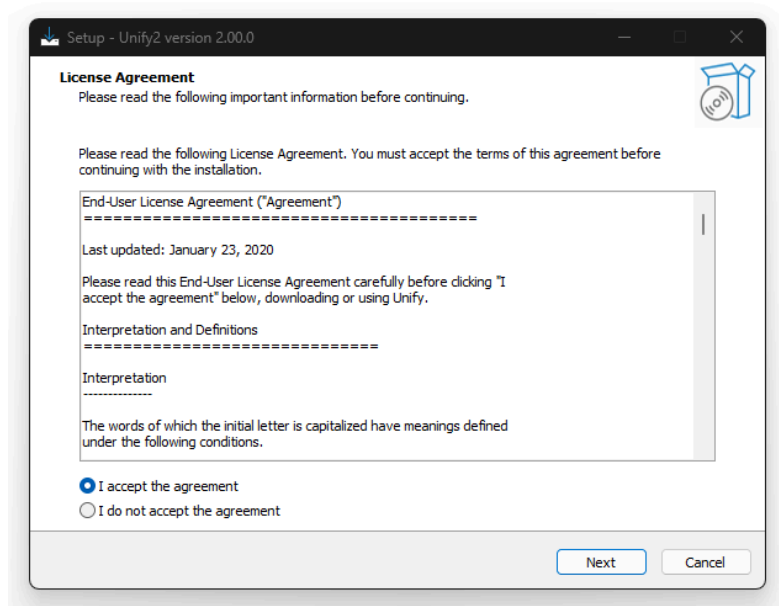
Please read the READ ME file before starting the installation. It may contain up-to-the-minute details that affect how you do the installation.

Double click the downloaded installer file:

In Mac OS this will be Unify\_2.00.0\_signed.pkg (or later build number)

In Windows OS this will be UnifySetup\_2.00.0.exe (or later build number)

You will be presented with a code signed installer – please accept the agreement and follow the steps.



In Windows OS, the plugin will be installed by default to the c:\Program Files\Common Files\VST3\ folder.

In Mac OS, the plugin will go to the Library\Audio\Plugins\ default folders for VST3 and Components.

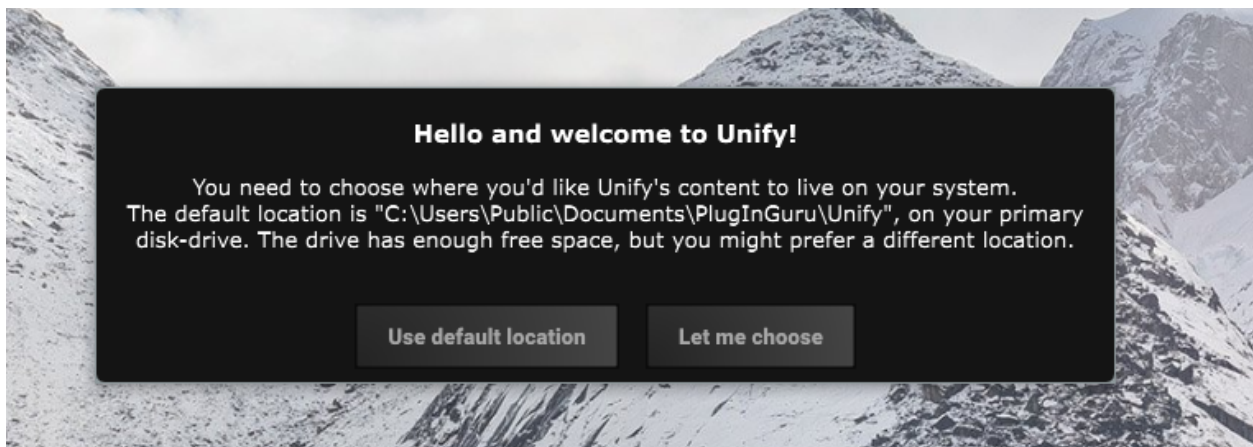
## Activate and set up Unify 2

### Run the standalone app

Once the installer has completed, please run the standalone version of Unify 2.

In Windows this is located in C:\Program Files\PlugInGuru\Unify 2\Unify 2.exe  
In Mac OS, this is located in Applications\PlugInGuru\Unify 2\Unify.app

On first launch, if this is your first time using Unify or you don't already have Unify 1.x installed, the first thing that will appear is a popup asking you to pick the default content location:



As Unify libraries can get quite large we suggest to use a fast secondary disk with lots of storage space via the let me choose option rather than the system disk – otherwise use the default option which will use the shared users / public documents location.

### Note for existing Unify 1.x users

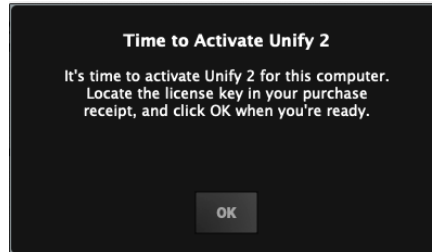
You can point Unify 2 to your existing Unify 1 content library – please be aware that any patches saved with Unify 2 cannot be used within Unify 1.

All content made for Unify 1 will load properly inside of Unify 2 – you may need to do a full database re-scan to update the patch list in Unify 2.

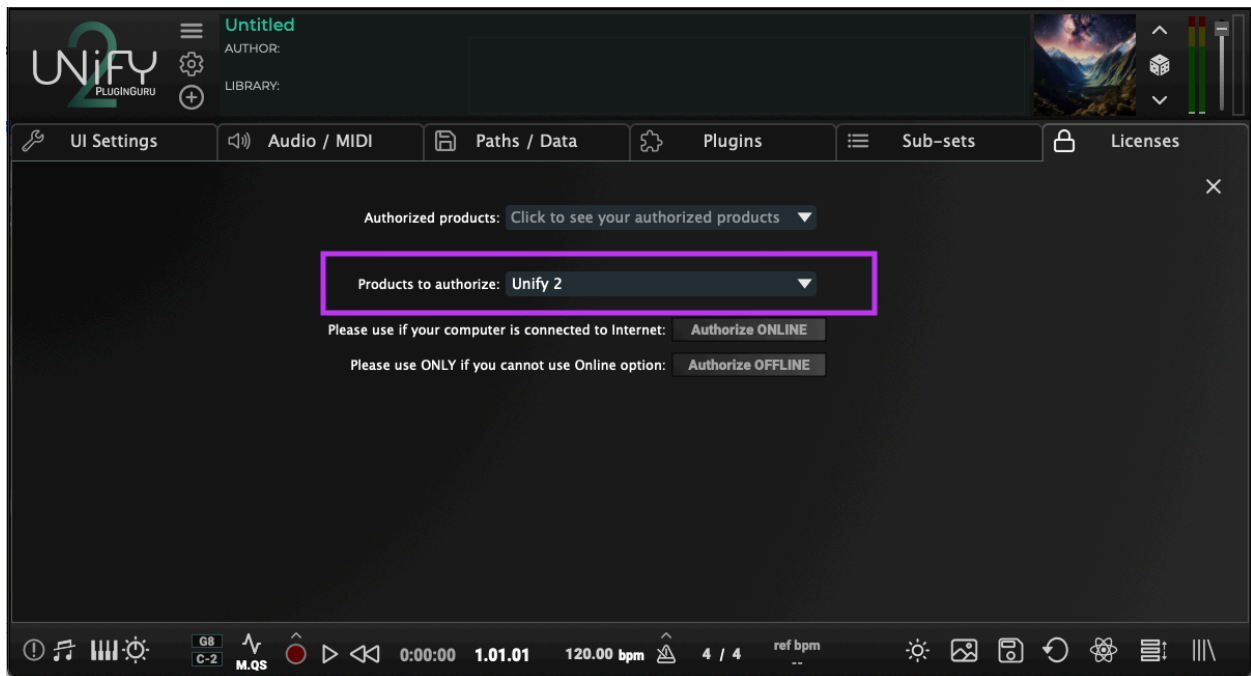
When Unify 2 is run for the first time, it will copy any unlocked product licenses from your Unify 1 user settings file, into the Unify 2 licenses file – so you can start using those sound libraries immediately.

## Activate your license

The next step after choosing the content folder location is product activation:  
You will see a popup dialog as follows:



Check the key in your email, or log into your PlugInGuru account to get your product key. Use the license page to do an online activation. If you are not able to perform an online activation, there is an offline process – you can contact support if you need help with this:



Your License Key is a string of 32 letters and numbers (only the digits 0-9 and letters a-f will appear), e.g. e4cdea654922e3a6320c3984631fe52c.

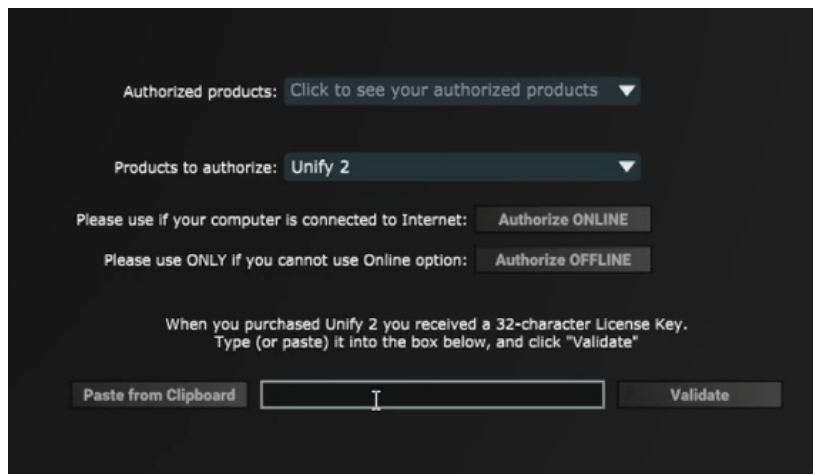
Locate your License Key now (either in the email receipt sent to you or by logging into the PlugInGuru.com website. If you can, select and copy it to your computer's clipboard, ready to paste into Unify.

If your computer has an Internet connection, click the “Authorize ONLINE” button.

If this is a studio computer which is not connected to the Internet (or if you try the online procedure and it fails for some reason), click the “Authorize OFFLINE” button.

### Online authorization (recommended)

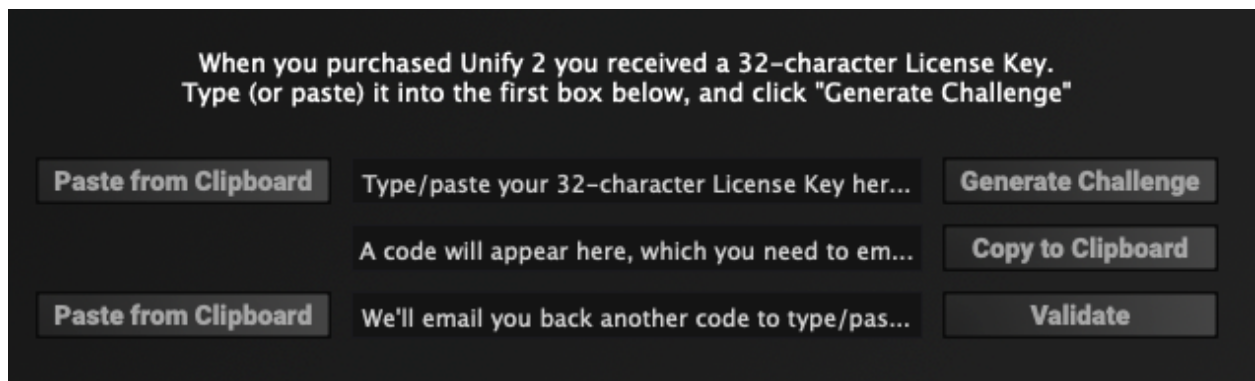
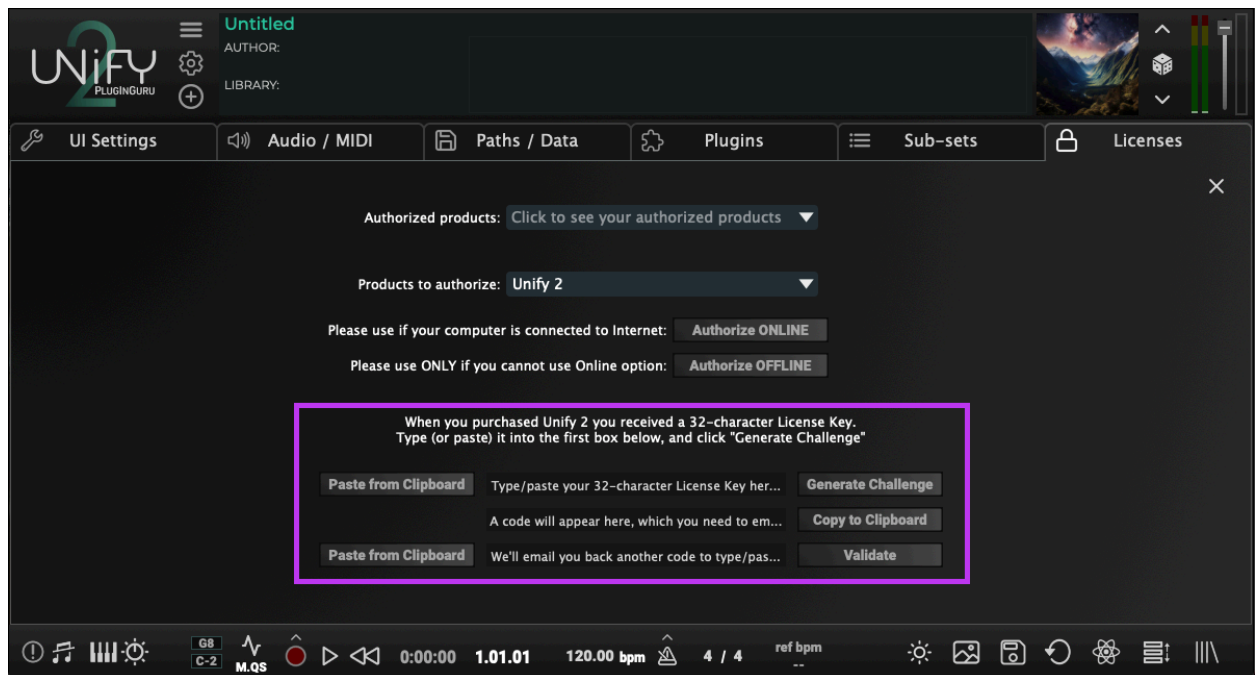
When you click the “Authorize ONLINE” button, the following dialog will appear:



- Locate your 32-character License Key, select it, and use Ctrl-C (Windows) or Cmd-C (Mac) to copy it to your computer's clipboard.
- Click the “Paste from Clipboard” button in the Unify GUI, to put the License Key into the edit box, then click the “Validate” button.
- If successful, within a few seconds you'll be rewarded with a message in green text, saying “Valid license code accepted for Unify (Standard Edition). Thank you.”
- If unsuccessful, follow the onscreen instructions. Contact [UnifySupport@PlugInGuru.com](mailto:UnifySupport@PlugInGuru.com) if you need help.

## Offline authorization

When you click the “Authorize OFFLINE” button, the following will appear:



- Locate your 32-character License Key, select it, and use Ctrl-C (Windows) or Cmd-C (Mac) to copy it to your computer's clipboard.
- Click the FIRST “Paste from Clipboard” button in the Unify GUI, to put the License Key into the TOP edit box, then click the “Validate” button.
- Click the “Generate Challenge” button. A lengthy code (16 groups of four characters, separated by hyphens) will appear in the MIDDLE box.
- Using whatever method works best for you (e.g. copy/paste into a document on a USB drive), you need to put this code in an email message and send it to [UnifySupport@PlugInGuru.com](mailto:UnifySupport@PlugInGuru.com).

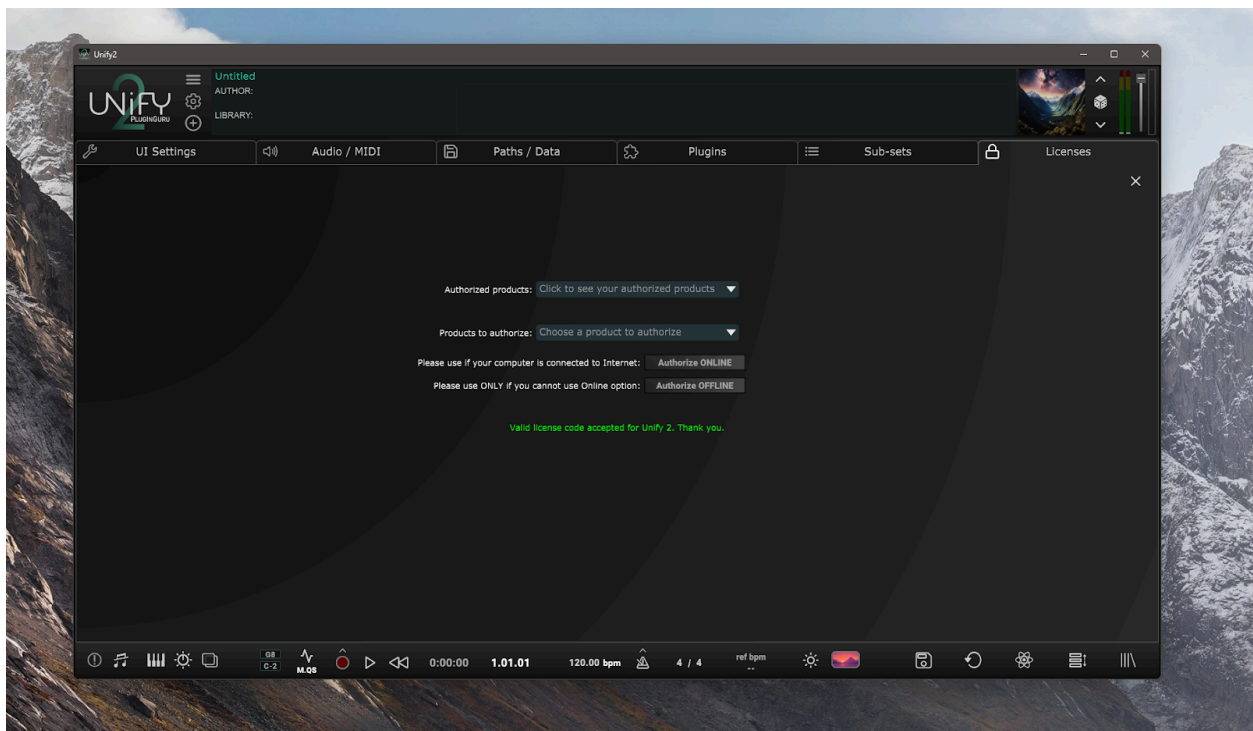
At this point, you'll have to wait until you get a reply with the “response code” you need to enter into the bottom edit box. If you want, you can close Unify.

When you do receive the code, restart Unify and start again from the first step above.

- Type or paste the “response code” into the BOTTOM edit box, and click the “Validate” button.
- If successful, you'll be rewarded with a message in green text, saying “Valid license code accepted for Unify (Standard Edition). Thank you.”
- If not, whatever message appears should guide you in what to do next. Contact [UnifySupport@PlugInGuru.com](mailto:UnifySupport@PlugInGuru.com) if you need help.

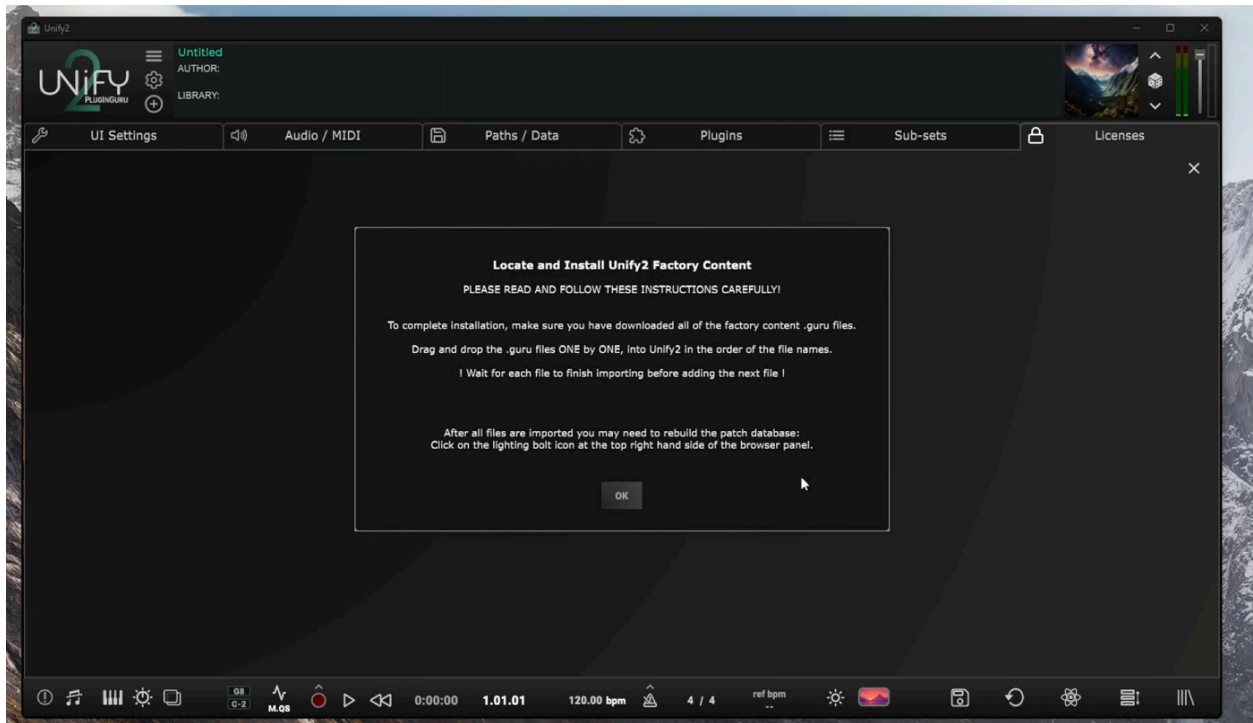
## Activation completed

You should see a green ‘valid license code’ message after correctly entering your product key

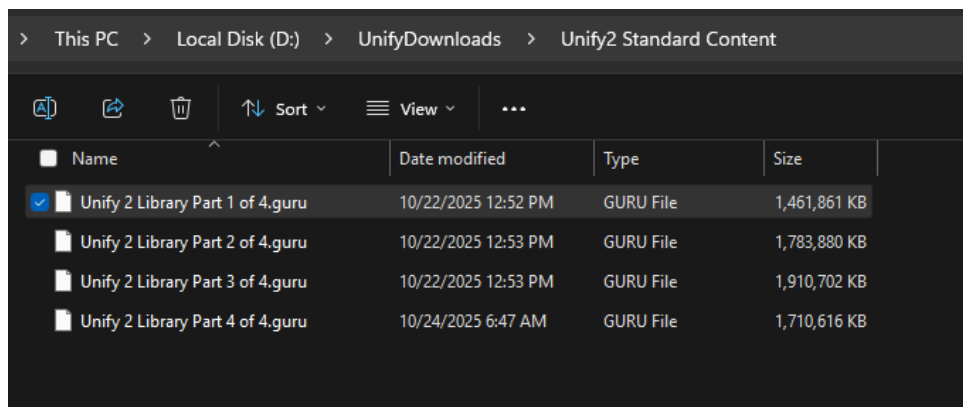


# Install Factory Content

Whether you choose online or offline licensing, as soon as you see the “code accepted” message, another message window will pop up, this time advising you to locate the *Unify Factory Content.guru* files, and get ready to drag its icon into the Unify window:

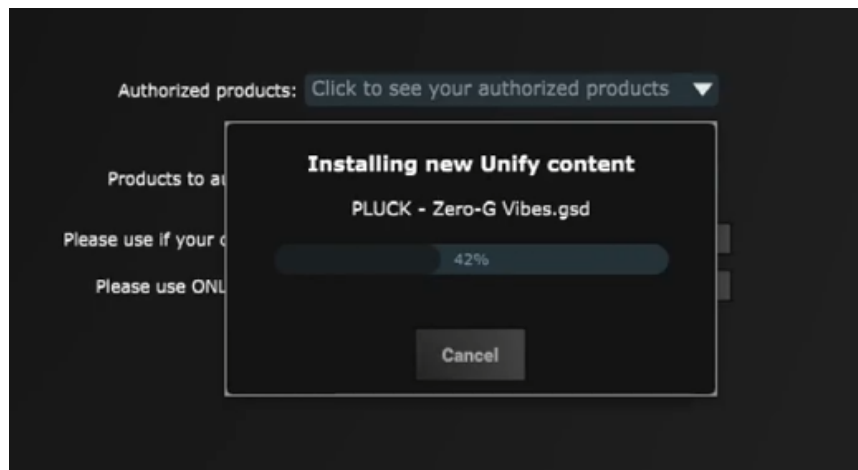
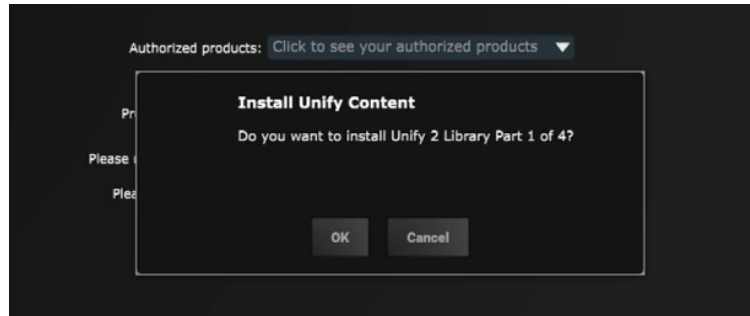


Locate the factory content files that you downloaded.  
There should be 4 files.



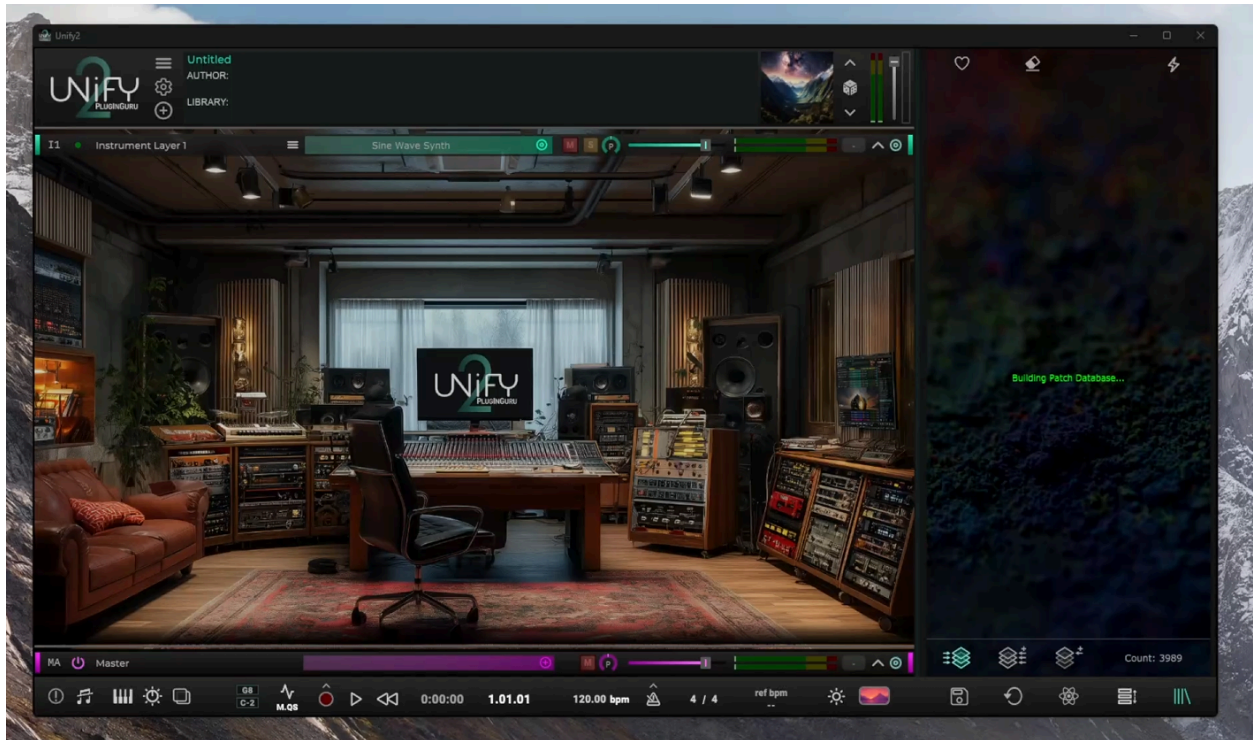
Drag and drop the files, one file at a time, in numerical order – as they are named, into the user interface of Unify 2.

Wait for each file to complete installing before moving on to the next file:



After the last .guru file has been imported, the user interface will drop back to the main screen of the interface and will begin re-scanning the patch list.

The patch browser will re-scan all the newly imported sound files – when complete it will show the finished list.



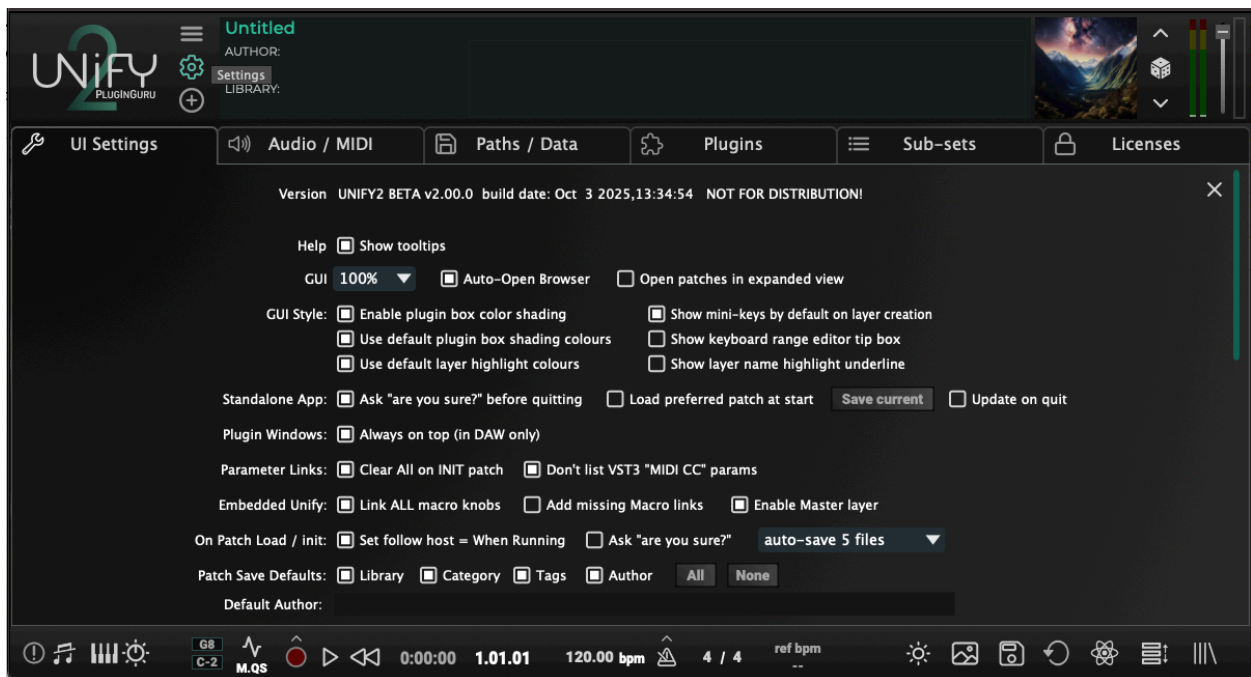
## Final Set-up steps

Before you can begin using Unify, you'll need to tell it about your audio and MIDI hardware.

Locate the Settings icon (it looks like a gear), and click on it; it's located to the right of the Unify 2 logo at the top right of the UI screen.

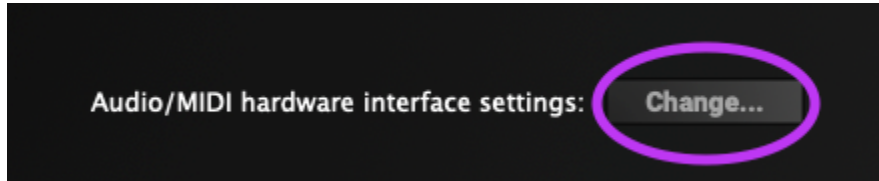
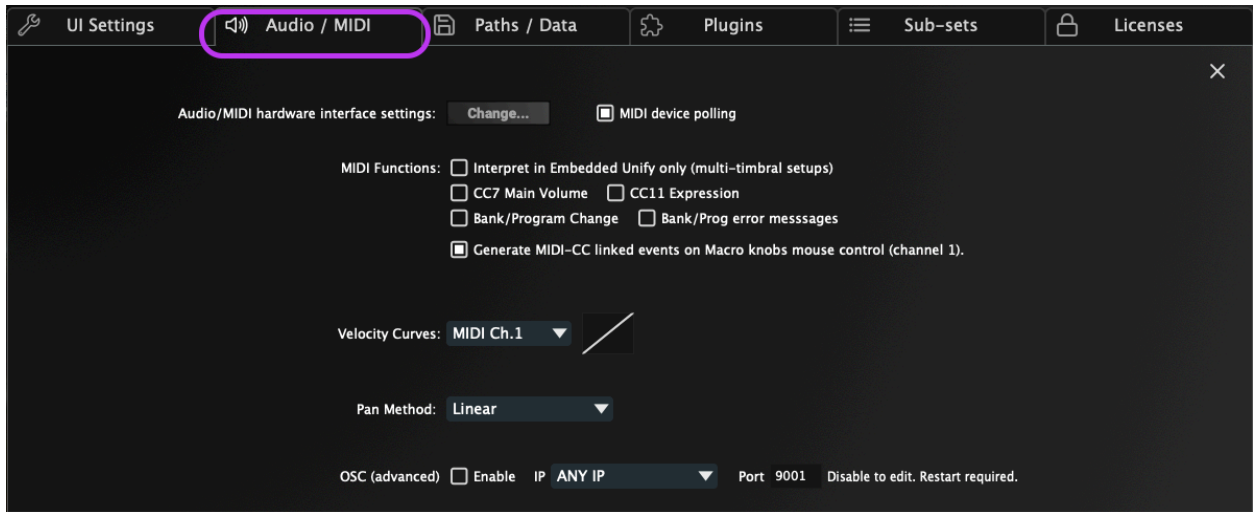


This will take you to Unify's *Settings view*, which looks like this:

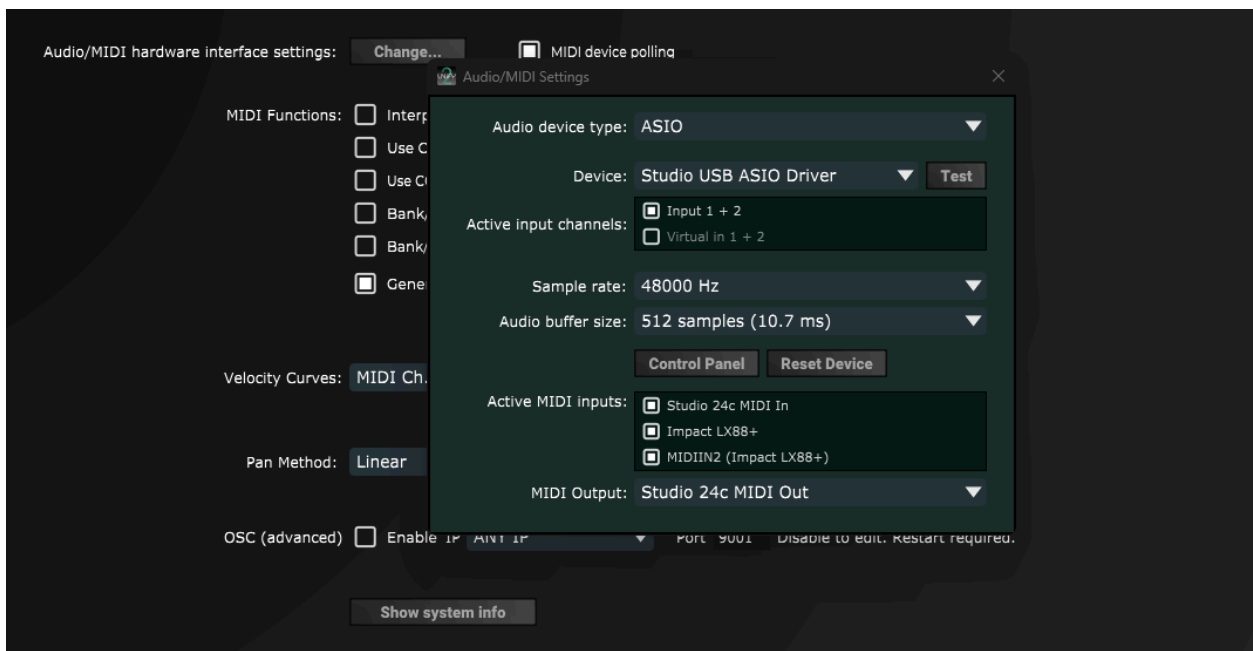


You will need to navigate to the "Audio / MIDI" section of the settings menu by clicking the "Audio / MIDI" tab in the Settings Title bar.

Click the “Change” button to set up your Audio/MIDI devices for Unify:



This opens the Audio/MIDI Settings window, which will look *something* like this; the actual appearance will vary, depending on your computer's operating system and what kinds of audio and MIDI hardware you have:



Make appropriate selections for your system, then close the window. (If you need help, contact [UnifySupport@PluginGuru.com](mailto:UnifySupport@PluginGuru.com).)

At this point, you should be able to play notes on your MIDI keyboard and hear sound (basic sine waves) from your speakers/headphones.

The selections you make in the Audio/MIDI Settings window are remembered, so you will normally only need to do this once.

## **What to do if you have problems**

If your Unify install doesn't go exactly as described above, you can contact us at [UnifySupport@PluginGuru.com](mailto:UnifySupport@PluginGuru.com).

# Unify 1 to Unify 2 upgrade - important notes

## Installation

When installing Unify 2, Unify 1 will NOT be erased - you can keep using Unify 1 as it is without any interference from Unify 2, however, there are some important caveats as noted in the following sections:

## Compatibility and shared library usage

When installing Unify 2 you can choose to point Unify 2 to your existing patch library. Unify 2 is 100% backward compatible with Unify 1 patches - all your existing libraries and saved patches should load without issue into Unify 2.

Unify 1 will not be able to load patches saved in Unify 2 - so you may decide that rather than keep a combined library it may be preferable to duplicate your Unify 1 library contents and use that duplicate copy for Unify 2, to avoid 'polluting' the Unify 1 library with Unify 2 patches.

## Licenses auto copy - Unify 1 to Unify 2

When you first run Unify 2, it will check for existing Unify 1 licenses. It will copy the licenses into the Unify 2 licenses storage file, ensuring that you can use Unify 1 libraries in Unify 2, if they were previously authorized on the same computer..

Please note that if you make further purchases and authorise them within Unify 2, they are not automatically unlocked in Unify 1. You will need to re-authorise them in Unify 1 if you plan to use them in Unify 1.

Going forward, it's likely that all newly developed commercial patch libraries will be developed in Unify 2, so they will not be compatible with Unify 1.

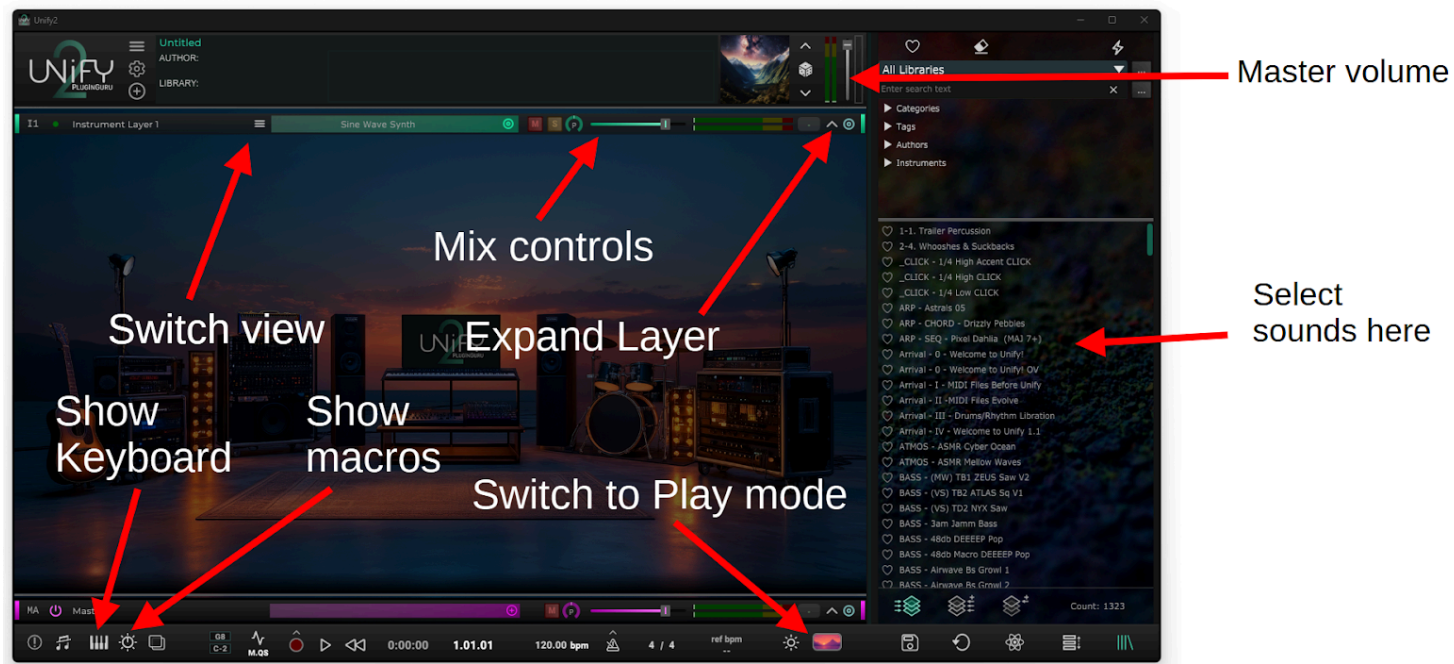
# UI Overview & Navigation

## Sections overview / Quick start

There are 3 main modes of the interface. Layers view, Play mode and Settings.

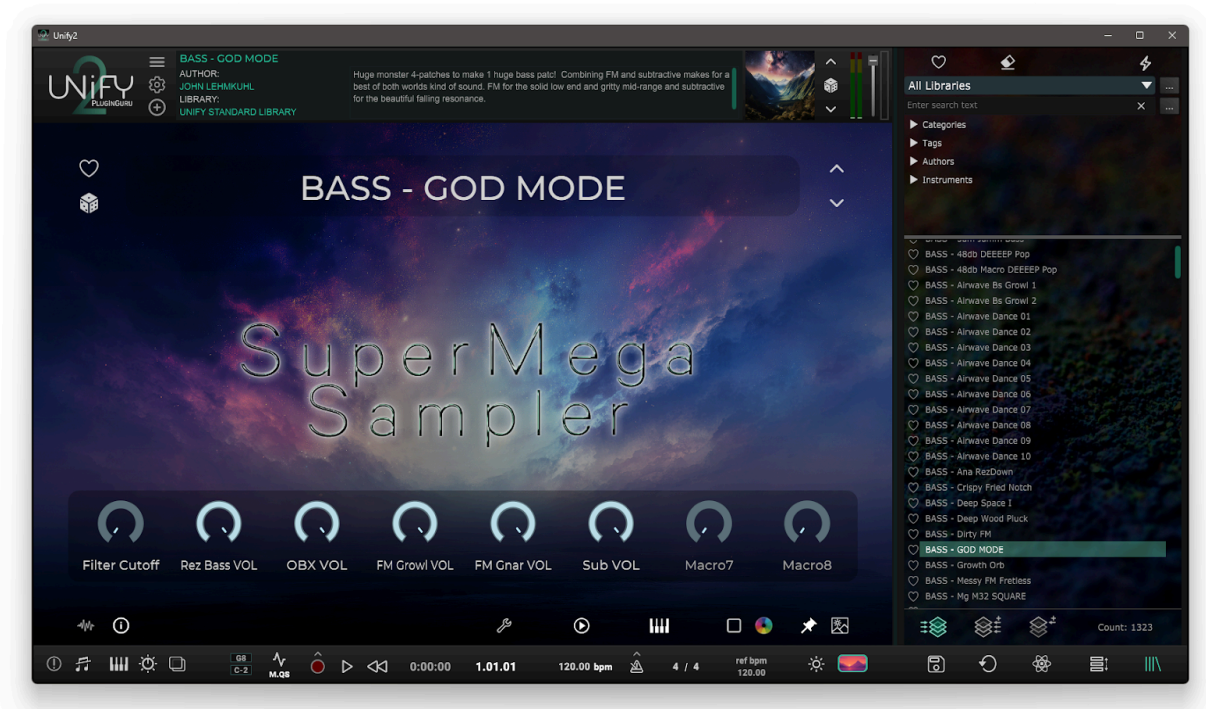
### Layers View - Intro

This is the Layers view, you can add new layers with the “+” button immediately to the right of the Unify 2 logo and beneath the Settings (Gear) icon.



## Play mode - Intro

This is the 'Play mode' view – it's designed to provide a minimum set of controls and let you focus on just playing the sounds.



The macro knobs seen on this page are the same knobs that are available in the layers view.

We'll go into more detail about this view later in the manual.

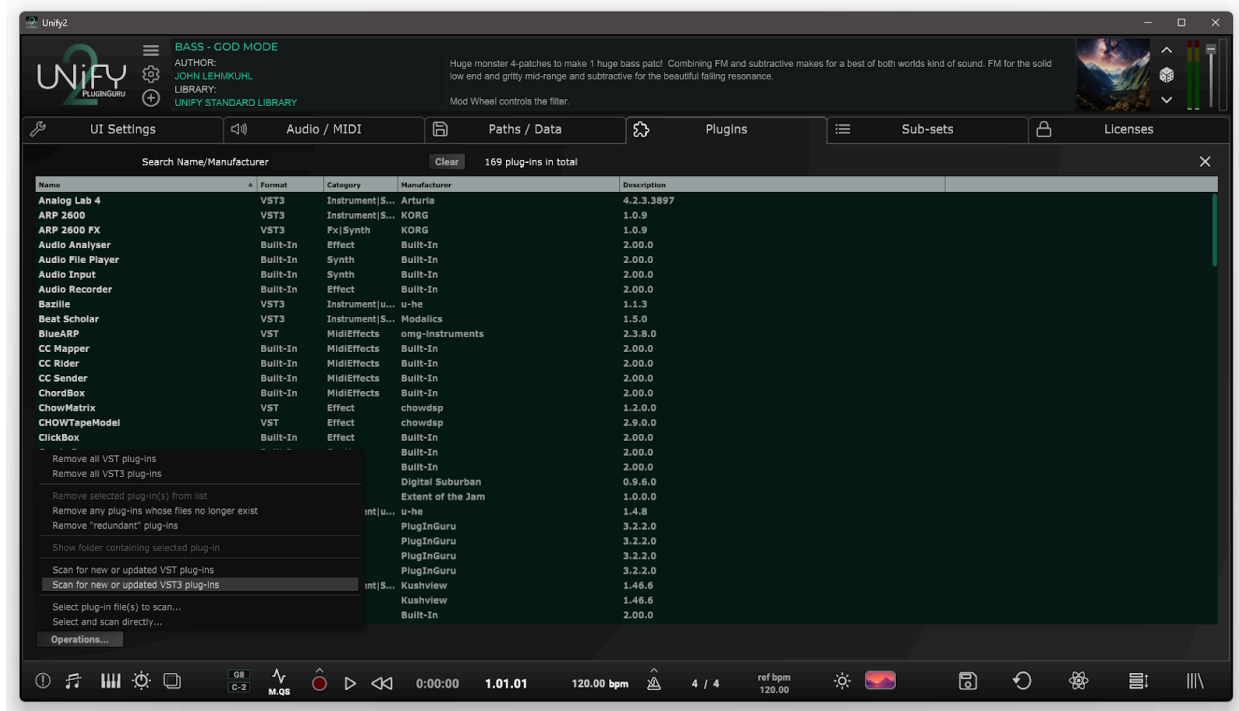
## Settings View - Intro

This is the 'Settings View', which has 6 tabs; UI Settings, Audio / MIDI, Paths / Data, Plugins, Sub-sets and Licenses.

Each tab has grouped related settings.

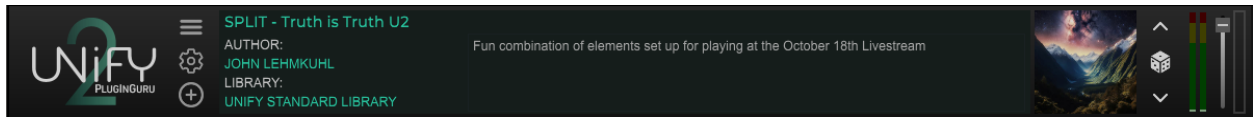
Although Unify 2 comes with built-in and bundled sound generator and processor plugins, one of the first things you should do is scan your own already installed plugins.

Do that in the Plugins tab:



A more detailed description of the Plugin scanning process is presented later in the Manual.

## Top bar / meta-data view



There are 3 main buttons to the right of the Unify 2 logo.

### Top - 'Burger' menu:

Here you can access information about Unify, some webpage links and some window management options.

### Top bar - Settings button

The settings button (cog icon in the middle) switches the interface into the settings tabs view.

### Top bar - Add layer button

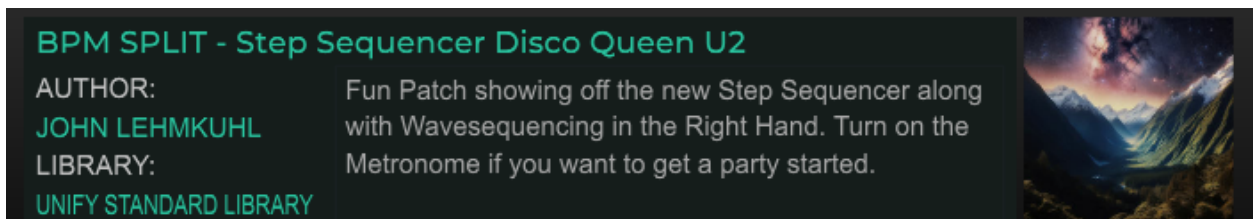
The add button (+ icon) is how you can add new layers to the layers panel.

There are some layer presets already available for some of the built in plugins, and otherwise you can choose to add an empty layer of either MIDI, Instrument or Aux bus type.

The master layer is always loaded and always available in the layers view.

The interface always has at least one instrument layer loaded when initialized.

### Top bar - Meta-data area



The meta data area shows details about the currently loaded patch.

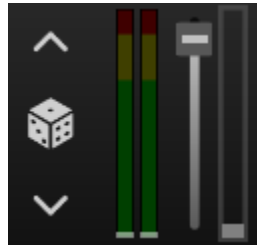
The image to the right is the artwork associated with a particular library.

If there is a larger image or a web link, clicking the image will open those in a separate viewer.

## Top bar - Patch navigation, master volume and CPU meter

To the right of the meta data area, the Preset browser navigation buttons allow you to move through the patch list or you can click the dice button to select a random patch within the currently filtered list.

To the right of those buttons is the master output volume meter and volume control slider.



To the right of the master volume slider is the CPU load indicator (actually it's audio processing load).

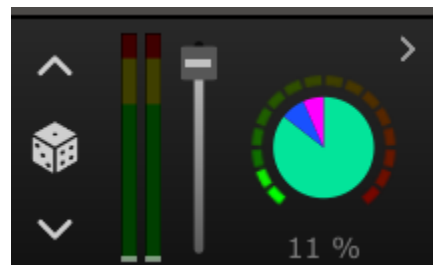
There are some options available by right clicking on the volume slider - in standalone mode you can set a default volume which will apply next time you open Unify 2.

If you click on the bar graph showing the current total load, the CPU meter will expand to show a pie chart with the proportional loads of the different layer types indicated by the different colours - matching the default layer colors.

Note that 'CPU load' in audio applications refers to the audio processing load on your system and is actually not the same as your actual CPU load. The load is the amount of time used as a percentage of the available time between each hardware request for an audio block.

Hover the mouse above the pie chart for more information about each section.

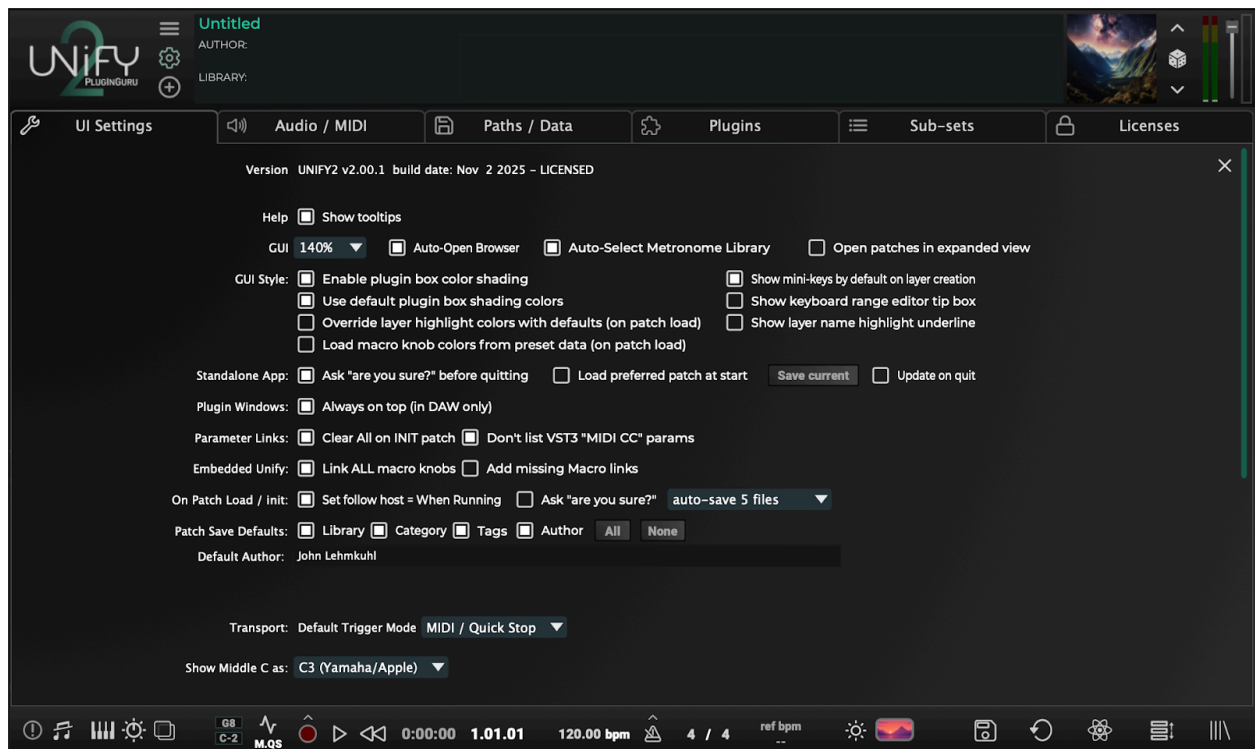
The indicator bars around the pie chart show the total audio load.



## Settings view - details

The settings of Unify 2 are organized in a panel with multiple tabs for each category of the application settings:

### UI Settings



Here you can set some 'default' behaviours of the User Interface (UI).

### GUI

When running on high resolution monitors, you might find it helpful to increase the GUI scaling factor - this will make the whole UI interface use larger fonts, larger icons and larger UI elements - at the expense of horizontal and vertical space for the layers view.

### Auto-Open browser:

When you launch Unify this makes sure the Preset browser panel is open.

Auto-select metronome library:

This option ensures that the Unify Metronome library is selected when opening the Metronome Unify instance.

Open patches in expanded view:

With this option checked, patches are loaded with the layers displayed in expanded mode.

## GUI style options

Enable plugin box colour shading:

If you un-check this checkbox, then all the different effect boxes for each layer will change to the current layer color.

Use default plugin box color shading:

By default, each type of plugin box has a different color; Unify 2 pre-defined colors.

If you uncheck this option, these colors will match the default color set for each layer type (via the Layer Color selector).

MIDI boxes - MIDI layer default colour.

Instrument boxes - Instrument layer default color.

Effect boxes - Aux layer default color.

Override layer highlight colours with defaults (on patch load):

This setting means that colors saved with patches are ignored. You may prefer to always use your own defined default colors, or use colors that are defined by the patch/library author. This setting gives the user control over that.

Load macro knob colours from preset data (on patch load):

As with the previous option, you can prevent knob colors changing when patches are loaded. If you change the macro knob colors, those colors are saved with the patch, but are only applied on loading if this check box is enabled.

Show mini-keys by default on layer creation:

Some people might prefer to turn off the dynamic mini-keys display on the MIDI and Instrument layer keyboard zones editor view. If unchecked, the dynamic mini-keys display will not automatically appear.

Show keyboard range editor tip box:

The tool-tip box that appears is intended to remind new users of how to set the range/curves and show the current values as you are dragging the range and curves. You can turn it off here.

Show layer name highlight underline:

Under each layer name there is a highlight line - you can turn it off if you prefer not to see it.

Standalone app options:

Are you sure:

Opens a dialog box that forces the user to confirm they wish to quit Unify. This can be helpful to prevent accidentally closing the app.

Load preferred patch at start:

Click the 'Save current' button, to set the current patch to the default patch at startup.

Update on quit:

Check this option to set the 'preferred patch' as the current state of the app when it's closed.

Plugin windows:

Selecting the Always on top setting tries to keep plugin windows to the front. This does not currently work for all DAWs.

Parameter links:

Clear All on INIT patch:

When this option is checked, when a patch is initialized all the macro knob parameter links are cleared, including to the Unify controls.

Don't list VST3 "MIDI CC" params:

Hides plugin VST3 MIDI CC parameters in the links editor.

Embedded Unify:

Link ALL macro knobs:

On opening new embedded instances of Unify, the macro knobs will follow the movements of the knobs in the main/top level Unify instance.

Add missing Macro links:

Add missing macro links when adding embedded Unify instances.

This will load the saved Macro Knob assignments to the 32 Macro Knobs of each embedded Unify Layer you add. If the Patches you are loading are organized in a similar fashion this can be a huge time saver, if they are very different from each other, you might not get the results you expect, but we created this option for cases where you have set up the Unify patches ahead of

time to have similar assignments (using the same knob for filter cutoff, for example between numerous patches you plan to layer together).

On patch load / init:

Set follow host = when running:

This is the default for plugin mode - the transport will follow the host/DAW transport when it is running, and run independently when the host transport is not running.

Ask "Are you sure":

If you made any changes to the current patch, this option will make a dialog appear to confirm the current patch should be auto-saved before replacing it.

Auto-save options:

Set the number of patch files that are saved in the undo history - each time you load a new patch the prior state is saved in the temporary undo buffer.

Patch save defaults:

These options auto-fill the default entries when opening the patch save dialog.

Default author:

You can set your name here, so that it's auto-populated when opening the patch save dialog.

Transport:

You can set the default trigger mode for when Unify is loaded/opened:

Trigger mode options are

- Manual start/stop - you have to click the play/stop buttons manually.
- MIDI quick stop - play/stop happens when notes are held or released.
- MIDI restart - playing notes starts the transport, playing again resets the position.
- MIDI latch - playing notes starts, manual stop is required to stop the transport.
- MIDI 1,2 4 etc bars - playing notes starts, on release notes the transport keeps running for the defined number of bars. This can be useful to keep a drum groove/metronome backing/arpeggiation running whilst releasing your hands from the keyboard for example - without those stopping immediately.

Show Middle C as:

There are three (3) display options:

C3 (Yamaha / Apple)

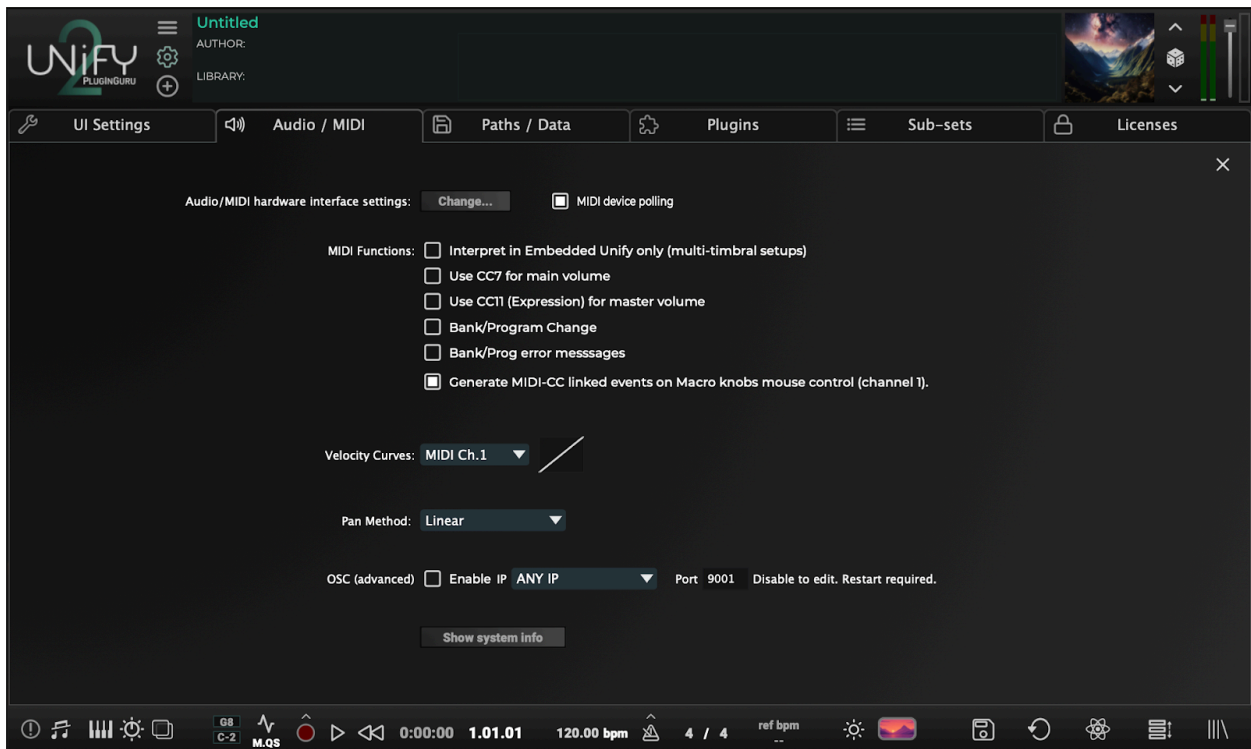
C4 (Roland)

C5 (FL Studio)

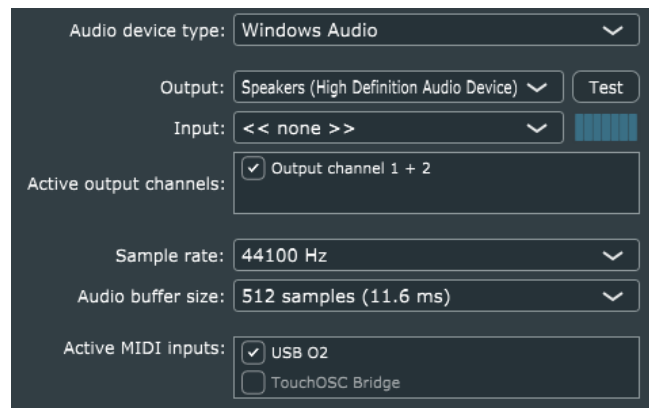
Link URLs:

Open in default browser - opens any meta-data image links in your computer's default web browser when you click the image.

Audio / MIDI:



This is where you set your preferred Audio and MIDI handling options. The most important option is your Audio/MIDI hardware interface settings. You can set your requirements by clicking the 'Change' button. A sample window is shown below, but this will be different for each user.



To enable audio input and output on multiple channels in the Layers view, you need to enable multiple extra audio inputs and outputs - in Windows this is usually only available if you select an ASIO driver/device in the i/o list.

The MIDI device polling option will auto-connect all detected devices - turn that off if you want to manually select just one device.

You can set Unify to respond to some standard MIDI messages on this page - the bank/prog change controls are related to a text file where you can set specific patch names.

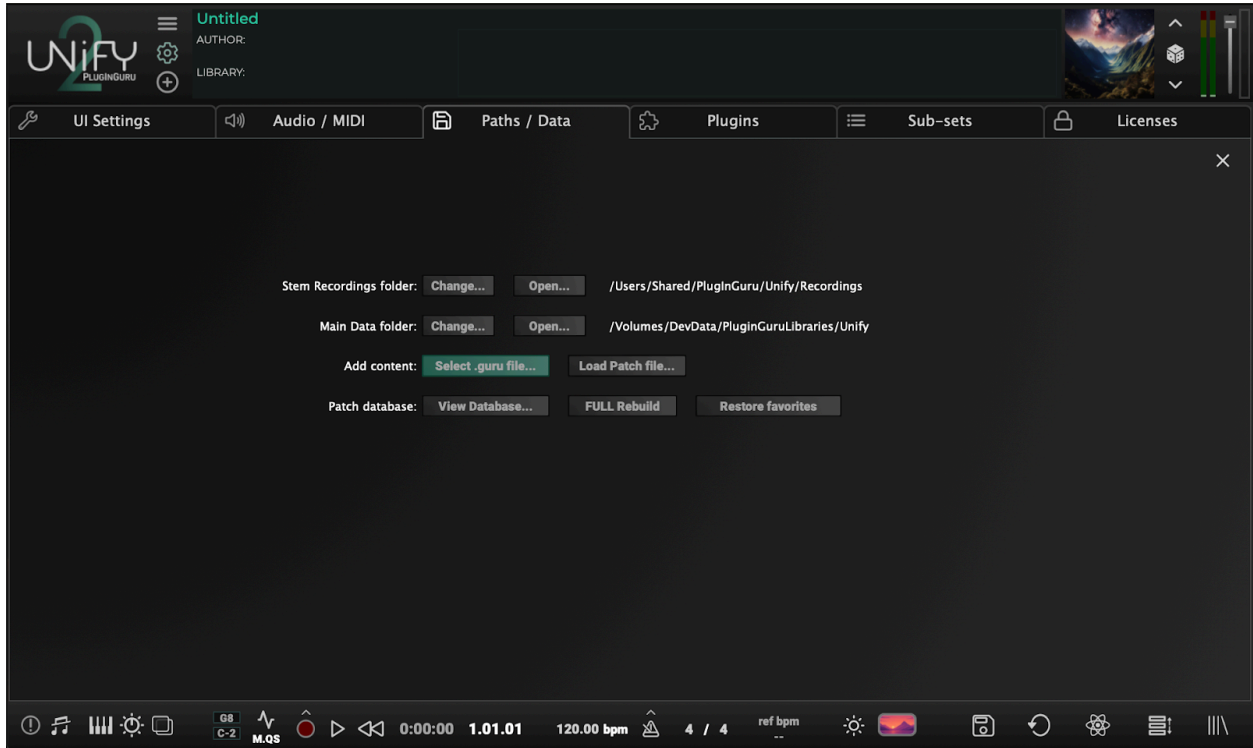
More information is in the original Unify 1 user manual which still applies to Unify 2:

<https://plugginguru.net/unify/manual/doku.php?id=midi-patch-select>

Similarly, OSC support is an advanced user subject that is fully explained in the online Unify 1 user manual:

<https://plugginguru.net/unify/manual/doku.php?id=osc-support>

## Paths / Data



The options on this page are self explanatory.

The Stems Recording Folder is the default location for Unify 2 recordings. Each time you make a new recording, a time-stamped recording folder is created in that directory where Audio and MIDI files are stored.

The Main Data Folder is the Unify 2 root data folder that holds all the layer presets, instrument presets, patch libraries and so on.

The Add Content options allow a user to import a .guru content file or single patch file via the relevant buttons. You can also just drag and drop these files onto the Unify UI and they will be imported.

There are three (3) Patch Database options.

The 'View Database' button opens a panel to show all the details of the current patch database This can be useful for checking if a patch import was done correctly.

The FULL Rebuild button deletes the existing Patch database and performs a complete rebuild rather than just adding any new patches found.

Restore favorites - the favorite patches list (normally held inside the database) is backed up into the Unify 2 user settings area, this button allows you to restore from that file if for some reason you lost them (e.g you accidentally deleted your full-database/content folders).

The restore favorites option can be very handy in a situation where you need to do a full-database re-build - e.g. if for some reason there are duplicate entries in the patch list that need clearing out.

Normally the favourites are backed up during the full re-build, but in case something goes wrong with that, Unify 2 is now keeping a separate back up copy which gets updated every time you close the interface or do a patch scan - so even if the database got completely wiped out, the backup list should help you recover the marked favorites.

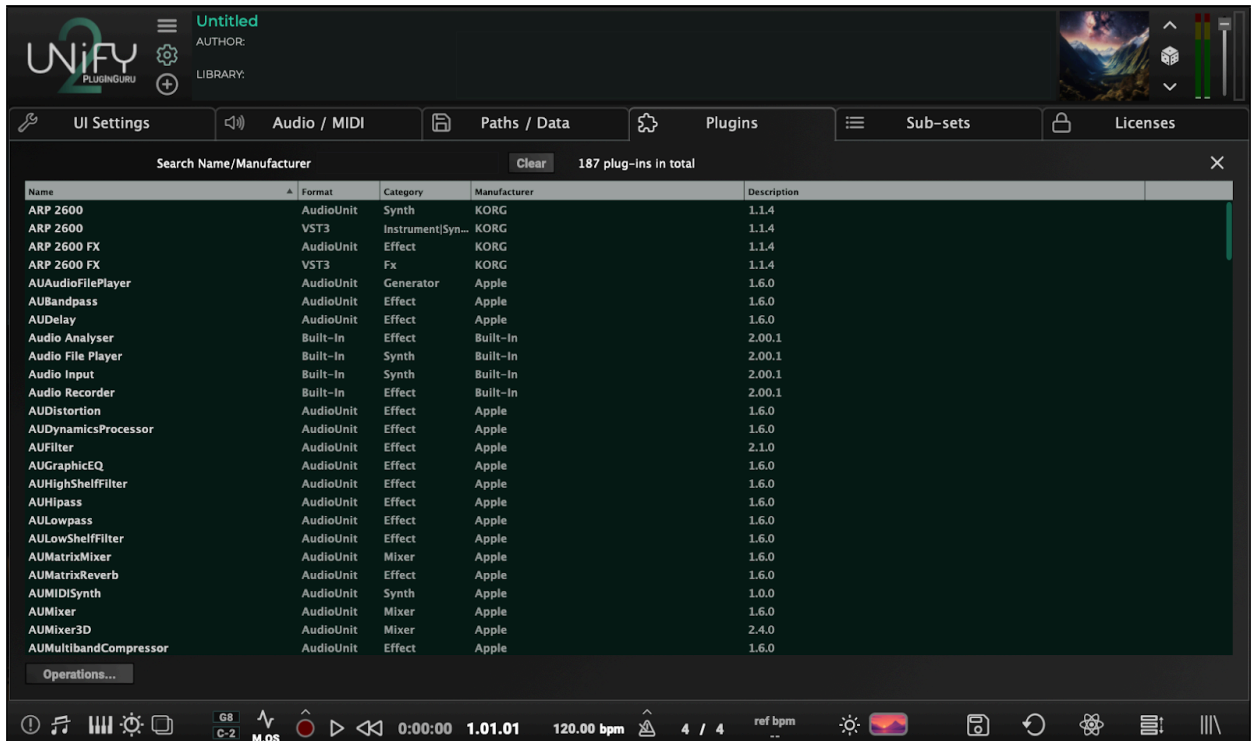
## Plugins

The Plugins tab displays the list of 3rd party plugins that have been registered with Unify.

You can filter the list by typing into the search box - to check if a plugin was added or not.

Right clicking on entries in the list allows you to remove individual plugins.

The plugins listed here are the plugins that will be available in the layer plugin box selection menus.



Before you can use any of your own plug-ins with Unify, they must be registered in Unify's plug-in database.

You can scan your system for available plug-ins, or you can register plug-ins individually.

## Scanning your system for plug-ins

You can also have Unify scan your system for available plug-ins.

Note that when scanning for plug-ins, Unify actually has to try to load each one.

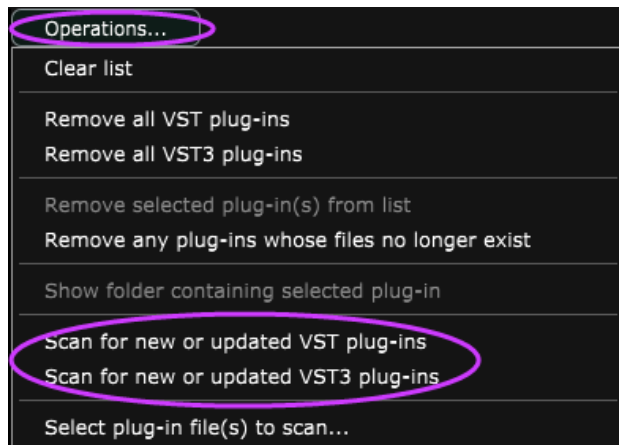
This will trigger any copy-protection mechanisms a plugin uses.

We therefore recommend the following tips:

- Scan using the stand-alone Unify app, not in your DAW.
- Avoid scanning your entire system if you can. It's perfectly OK to scan just one or a few folders at a time, and you can always come back another day to scan some more.
- Stay near your computer, so you can be ready to deal with e.g. pop-up authorization dialogs.
- Plug in any copy-protection dongles you have before scanning.

To scan your system for available plug-ins, begin by clicking the “Operations...” button at the bottom-left corner of the plug-ins view. This opens a small menu with several options, most of which are self-explanatory.

The scan options are at the bottom:

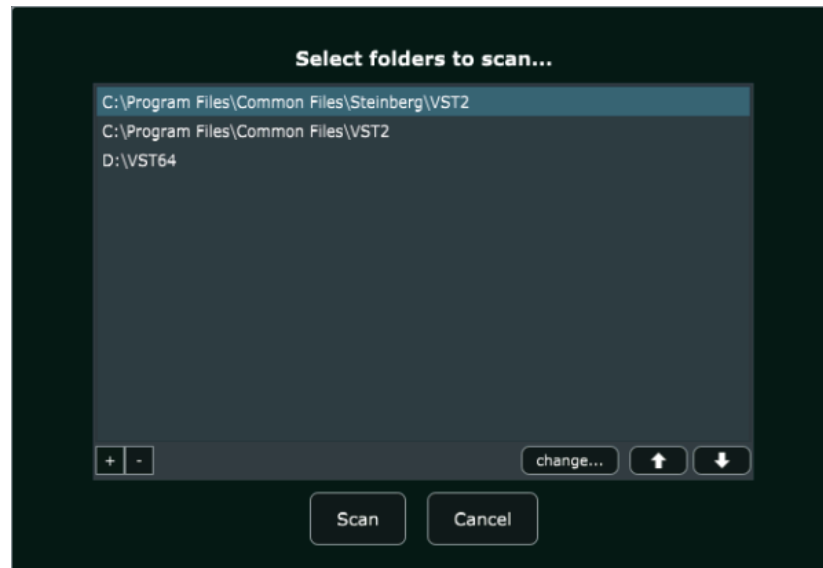


Note the above screenshot was captured on a Windows system, which only supports VST and VST3 plug-ins.

On a Mac, you would see a third option to scan for Audio Unit plug-ins.

Select the appropriate menu option for whichever plug-in format you want to scan for.

If you choose to scan for Audio Unit plug-ins (on a Mac), the scan will begin right away. If you choose to scan VST or VST3 plug-ins, a dialog similar to the following will appear, asking you to choose which folders to look in:



The VST and VST3 standards do not specify exactly where plug-ins ought to be located - although VST3 plugins are usually found in the Program Files\Common Files\ folder. Many systems (especially Windows systems) may use quite a few different plug-in folders.

It's up to you to know where these are, and how to specify the paths to them, just as you do for working with plug-ins in your DAW.

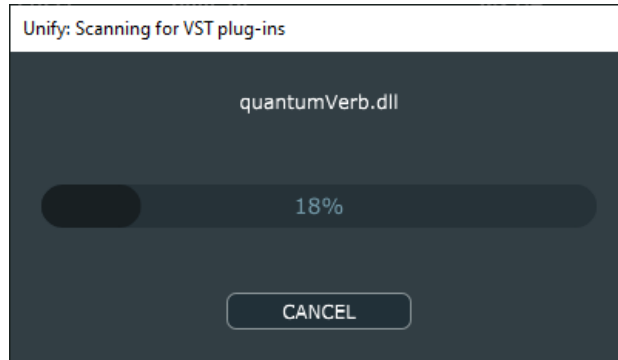
The “+” button at the bottom-left allows you to add another path to the list.

You can click any path in the list to select it, and either use the “-” button to remove it or the “Change...” button to change the path. (Once you have selected any path in the list, you can use the arrow buttons to move it up or down and thus change the order in which the folders are scanned.)

VST/VST3 plug-ins are often stored in nested structures of folders, e.g., on a Windows system, you might have most plug-ins in, say, C:\Program Files\Common Files\VST2, which also contains a sub-folder called Plogue Art et Technologie, Inc, containing all of that company's plug-ins.

If you select the top-level folder, Unify will scan into the sub-folder(s) as well, but you could also specify just the path to the sub-folder instead, and Unify would not attempt to scan all the others.

Once you have edited the folder list the way you want, click the “Scan” button to begin the scan. The dialog above will be replaced by another one showing the progress of the scan:



This new dialog is actually a separate “helper” program called PlugScanner.

You can drag the PlugScanner dialog to a different location on your screen, and continue using Unify normally even while it's running.

The “CANCEL” button in the PlugScanner dialog allows you to stop the scan gracefully, e.g. if you decide you'd prefer to do the scan another time.

Back in the Known Plug-Ins view in Unify, a progress message in green text will appear at the bottom of the main window, and the “Operations” button will change to “CANCEL SCAN” in red. Unlike the CANCEL button in the PlugScanner window, this one actually shuts down PlugScanner forcefully.

Use it only if you think PlugScanner is completely stuck, but give it a bit of time—some plug-ins take surprisingly long to test, so you might see no apparent activity for a minute or more at times.

PlugScanner may CRASH during scanning. Unify will automatically restart it, and it should continue from where it left off. If you previously moved the PlugScanner window, you'll immediately know that it was re-started, because it will seem to jump back to where it was at first.

In a few rare cases, PlugScanner may get stuck on one particular plug-in, re-starting again and again. If this happens:

1. Cancel the scan
2. Locate the problem plug-in in the Mac Finder or Windows Explorer, and move it to some other folder
3. Start the scan again
4. Once you are able to scan all the way to the end, you can move the problem plug-in back where it was.
5. Try the drag/drop method to re-scan just that plug-in—this works in many cases.

Various pop-up windows may appear during scanning. This is why we advise staying near your computer, so you can click “OK” or whatever, to allow the scan to continue.

- On a Mac, you may see a pop-up saying that some plug-in needs to access information from your keychain, and you must enter your password to allow this. After entering your password, we advise that you click “Always Allow” instead of “Allow”, because you will probably have other plug-ins from the same vendor, which will also need this access.
- If you have any expired demo plug-ins on your system, you will probably see pop-ups reminding you that they are expired. You will usually need to click “OK” to continue.
- If any of your plug-ins requires a hardware dongle to work, you'll see a pop-up about that. You'll have to click “OK” to continue, and in some cases, the plug-in won't register properly, because it will be unable to start properly.

Once a scan eventually completes, you may see one or more items in red text at the bottom of the plug-in list, with an indication that they “appeared to be plug-ins” but couldn't be opened properly.

- In some cases, these are not plug-ins at all, but simply auxiliary files stored in the same folders. These can safely be ignored.
- Occasionally, you might see a file listed which you're pretty sure is a plug-in. Try the individual registration method (see top of this page) to re-scan just that file—this works in many cases.

### Registering individual plug-ins

You don't have to run a full scan for plug-ins.

You can register plug-ins individually in one of two ways:

1. Locate the plug-in file and drag/drop its icon into Unify's *plug-ins view*
2. Click the “Operations...” button at the bottom-left, and choose “Select plug-in(s) to scan”
  - A standard open-file dialog will appear
  - Locate and select the plug-in you want to register, and click “Open”

## Plug-in scanning issues and work-arounds:

Unify's plug-in scanning system is quite robust, but occasionally, it may report that a particular plug-in can't be loaded, even though it actually can.

When this happens, locate the plug-in in the Mac Finder or Windows Explorer and drag/drop its icon directly into the Known Plug-Ins view.

This bypasses the scanner, and may succeed in some cases where the scanner fails.

(If you are unable to drag/drop (some Windows set-ups block it), click the "Operations..." button at the bottom of the Known Plug-Ins view, choose "Select plug-in file(s) to load..." from the menu, and then locate the plug-in files by navigating in the usual way in the file-open dialog which appears.)

In a few rare cases, the PlugScanner helper app may get stuck on one particular plug-in, re-starting again and again - you can use the following strategy to find the problem plug-in:

Cancel the scan

Locate the problem plug-in in the Mac Finder or Windows Explorer, and move it to some other folder

Start the scan again

Once you are able to scan all the way to the end, you can move the problem plug-in back where it was.

Try the drag/drop method to re-scan just that plug-in—this works in many cases.

If you're not sure exactly which is the "problem" plug-in at step 2, you can find out as follows:

Watch the PlugScanner dialog to get a rough idea of which plug-in is the problem one. It scans in reverse alphabetical order, so you only need to be close.

Moving a group of plug-in files—from a little before, to a little after where PlugScanner keeps getting stuck—out of the plug-in folder.

Re-run the scan, to see if it can get past the problem plug-in.

Once the scan makes it to the end, move the plug-ins back into the original folder, one at a time and re-scanning until you find the one that makes PlugScanner crash.

Carry on from step 2 above

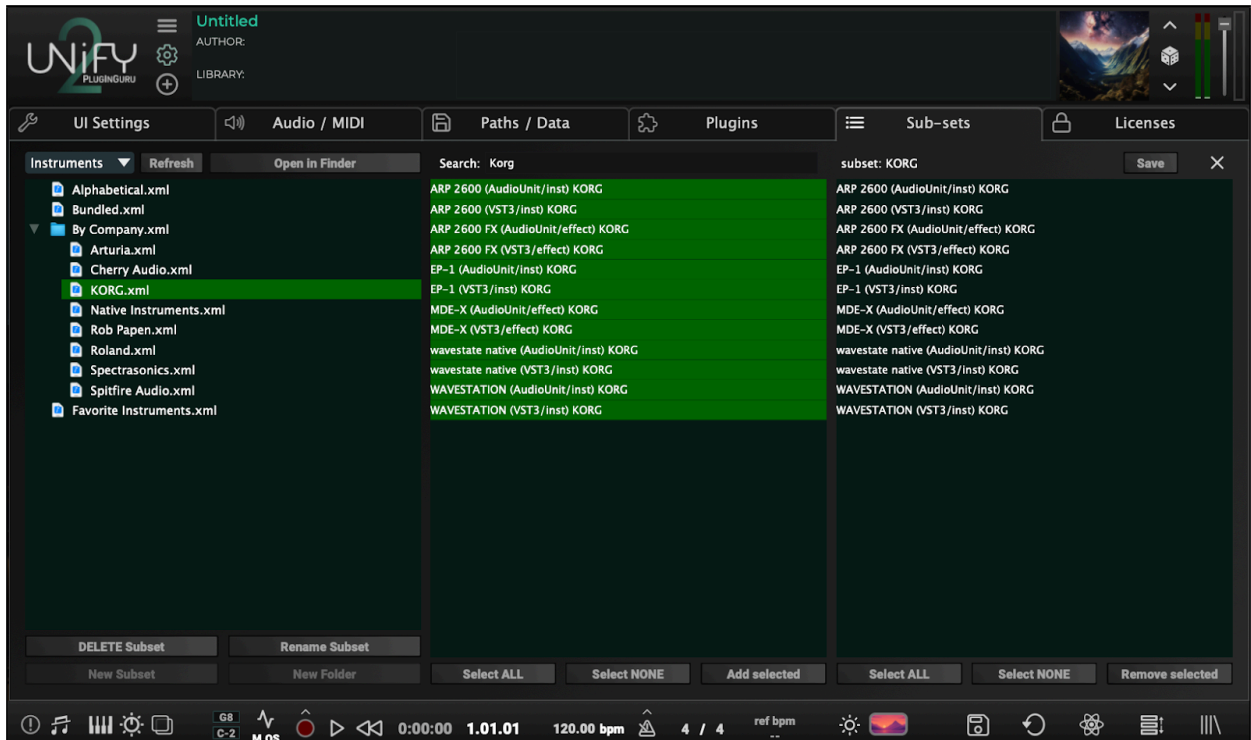
If you run into any problems with the plug-in scanner, we'd like to hear about them.

Any feedback you can provide COULD be the information we need to understand why something isn't working.

So email us at [unifysupport@plugginguru.com](mailto:unifysupport@plugginguru.com) if you have any information to share.

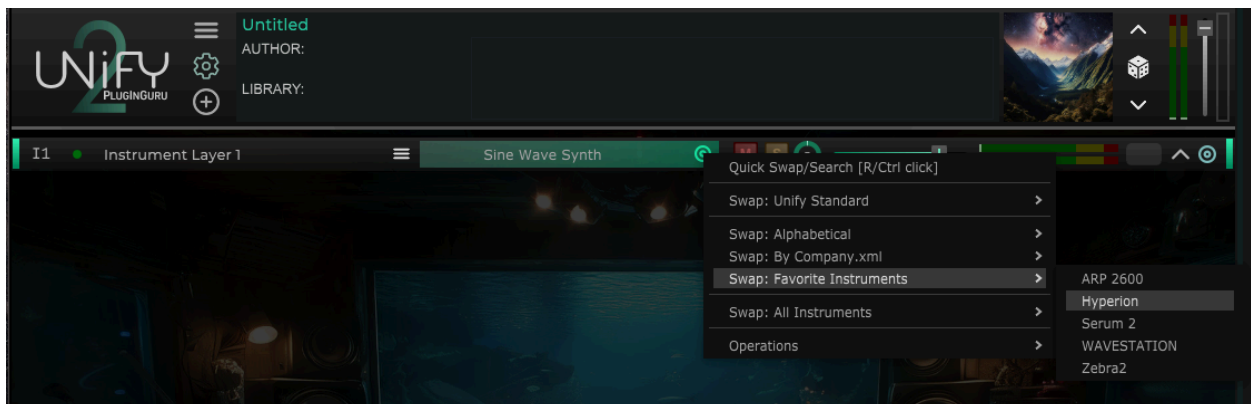
## Sub-sets

This is the subsets editor page in the settings tabs:



Plugin-subsets are a way to make it easy to find plugins from a particular manufacturer or your own specific list of favorite plugins.

When you create a sub-set it will appear in the layer plugin quick swap menus and allows quick access to sub-lists (sub-sets) of plugins avoiding the need to scroll through every single plugin installed on your system.



## Using the sub-set editor

The subset editor is split into 3 columns from left to right.

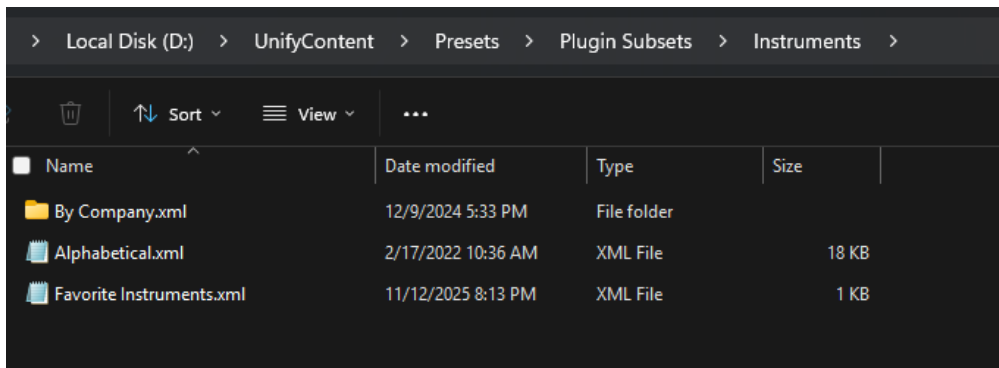
You can create subsets under 3 categories selectable in the 1st column of the sub sets editor page - Instruments, Audio Effects, MIDI Effects (select the type from the pull-down list).

The first column is where you define the structure of the menu that will appear after the 'Unify Standard' item in the swap menu.

Here you can create new named subsets (these are xml files) - simply click 'New Subset' to add to a selected sub folder.

These xml files are stored in your content folder inside the 'Presets/Plugin Subsets' folder.

Here for example is the Instruments subsets folder:



Highlight the newSubset.xml and click 'Rename Subset' to give it a custom name.

To add new plugins to the selected subset in the first column you pick plugins from the middle column:

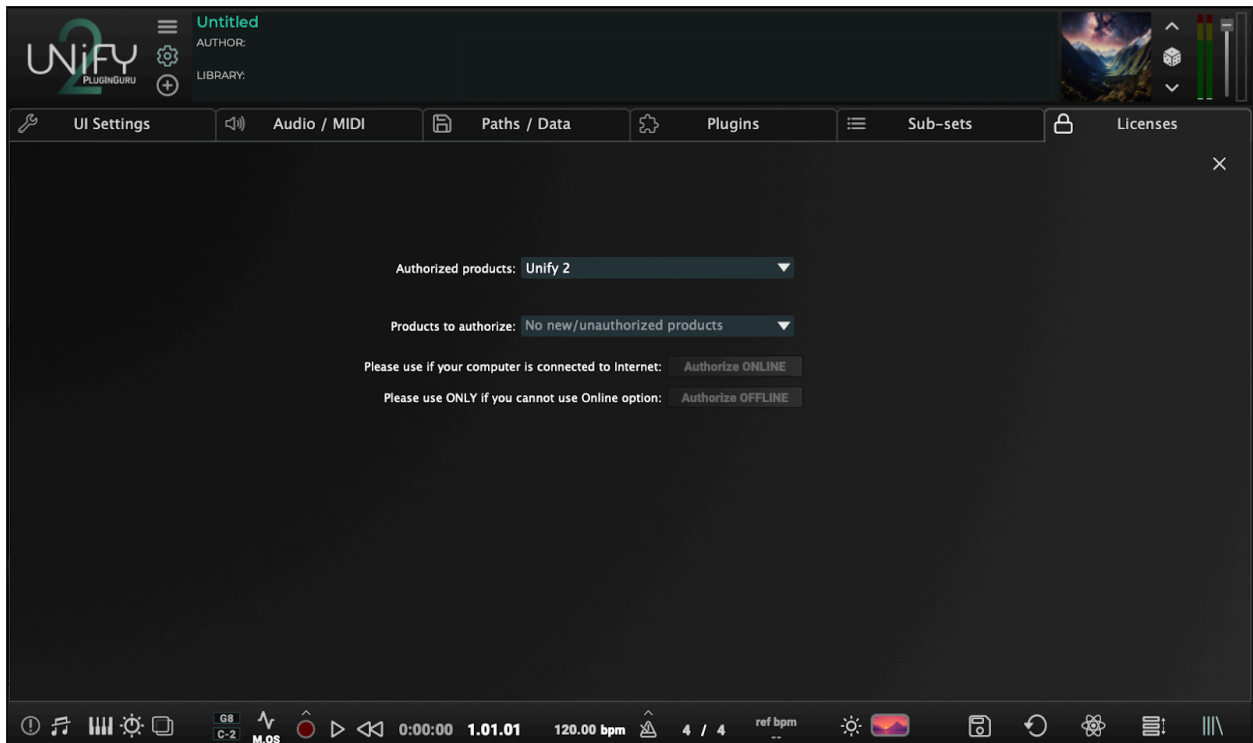
The middle column lists all the plugins that were found in the plugin scan.

You can select plugins one by one, or ctrl-click to select several, then click 'Add Selected' to move them into the 3rd / current subset column.

If you need to remove specific plugins select them here and click 'Remove selected'.

Once you are happy with the list for a sub-set, make sure to click the save button at the top of the sub-set column. After you've saved your subset, it will become visible in the plugin swap menus in the layers interface.

## Licenses



As previously described in the Setup/Quick Start - here is where you can view your current authorized products and authorize Unify 2 and any new licensed libraries.

After you drop a new .guru file with licensed content into Unify, it will appear under the 'Products to authorize' list.

Select the product and click the authorize online option - this is the best way to unlock the product.

For offline authorization, please check the details already described in the installation section at:

[Unify2UserGuide](#)

## Layers view - details

Unify creates composite sounds using a system of *Layers*, of which there are three distinct kinds:

1. **MIDI layers (MIDI)** host *MIDI-insert plug-ins*
2. **Instrument (INST)** layers host a chain of plug-ins as follows:
  - zero or more *MIDI-insert plug-ins* (which modify MIDI data)
  - exactly one *Instrument plug-in* (which produce stereo audio in response to incoming MIDI)
  - and a chain of zero or more *Audio-effect plug-ins* (which process stereo audio)
3. **Audio Effects layers (AUX)** contain only a chain of zero or more *Audio-effect plug-ins*
  - There is always exactly one **Master Effects** layer, into which the sounds from all other layers are mixed.
  - Some Unify patches may also use **Auxiliary Effects (AUX)** layers which are comparable to “aux bus” channels in a DAW or a traditional mixing console.



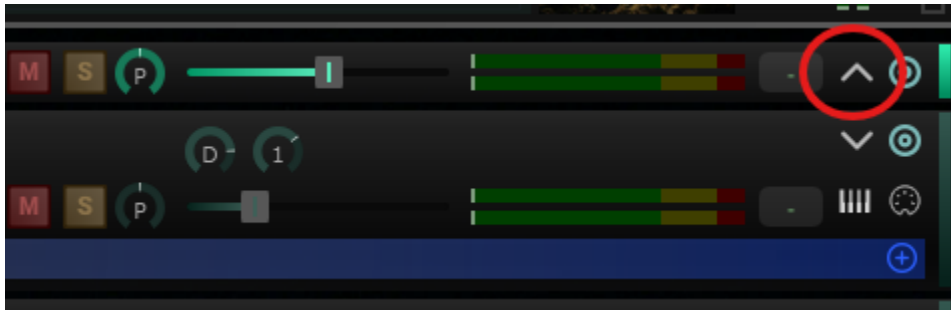
## Layers signal flow

MIDI and audio flows generally from top to bottom and left to right:

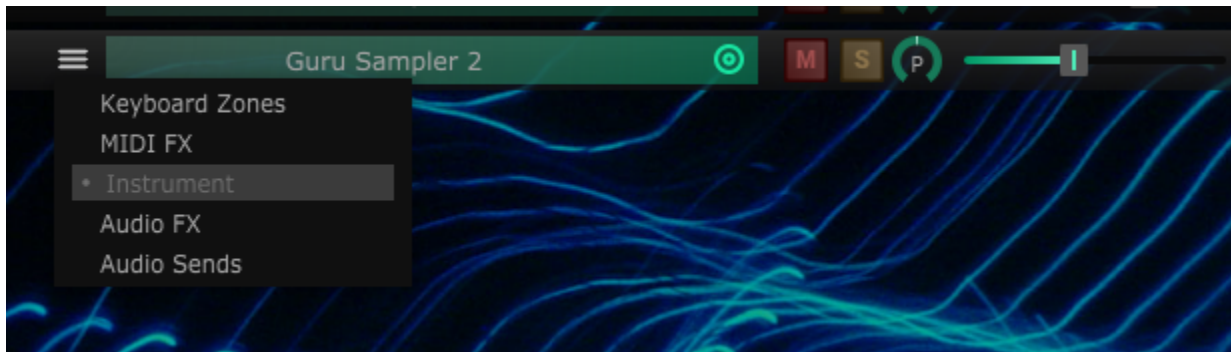
- Incoming MIDI data is sent through all MIDI layers at once.
  - Within each layer, MIDI flows from left to right through the chain of MIDI-insert effects
- Each INST layer can be set to receive MIDI either from the main input, or from the output of any MIDI layer
  - Within each INST layer, MIDI flows from left to right through any MIDI-insert effects, into the Instrument plug-in, and then the instrument's output audio flows from left to right through the chain of audio-insert effects
- Each AUX layer receives a **mix** of all INST layer outputs (set by the corresponding *aux send* knobs on each INST layer)
  - Within each AUX layer, audio flows from left to right through the chain of audio-insert effects
- The Master Effects layer receives a **mix** of all INST and AUX layers (set by the horizontal faders at the right of each layer)
  - The mixed audio flows from left to right through the chain of audio-insert effects
  - The processed output is sent to Unify's main stereo output.

## Layer controls

Layers can be individually minimized or maximized by clicking the chevron button at the right hand side of each layer - right clicking will minimize or maximize the whole group of related layers.

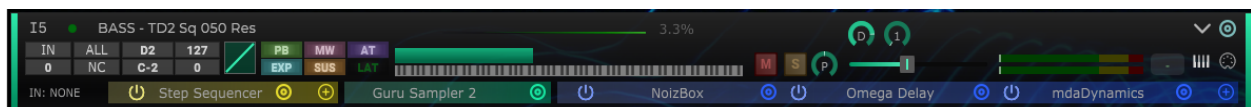


In the minimized mode you have access to a reduced set of controls, but on Instrument layers the central area can be switched via the Burger menu - you can switch between the Keyboard Zones, the MIDI FX, the Instrument, the Audio FX box and the Audio Sends box.



Right clicking (or holding the Shift key) and selecting an option will apply that option to all Instrument layers in the current view.

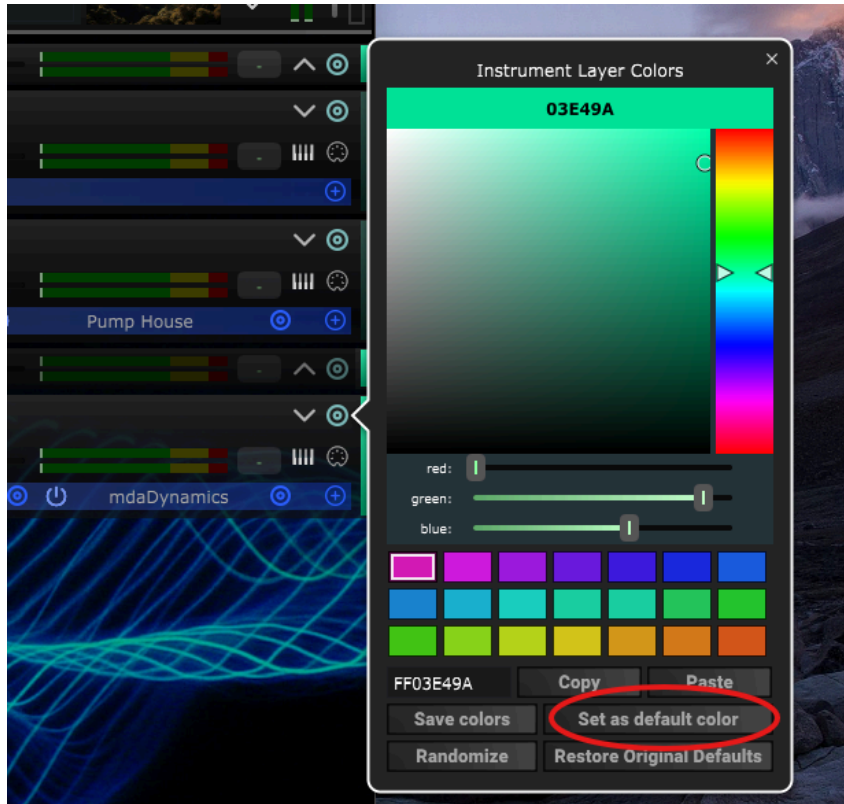
In the expanded mode, for the Instrument, Aux and Master layers, there is a dedicated row showing all the plugin boxes. When there isn't enough horizontal room for plugins, there will be a right/left chevron button you can click to scroll through the loaded plugins.



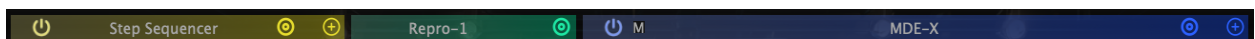
By default MIDI boxes/Layers are shaded Yellow, Instrument boxes/layers are Green, and FX boxes/layers are Blue. The Master bus layer's default color is purple.

Default Colour settings for layers can be adjusted in the Settings panel / UI settings tab, but this is still an experimental feature.

Layers can be set to custom colours via the colour menu of each layer, accessed by right-clicking on the Layers Operations (Target icon) button. The default colour for a layer type can be set for consequent layer additions or new patch creation here.



## Layer plugin boxes



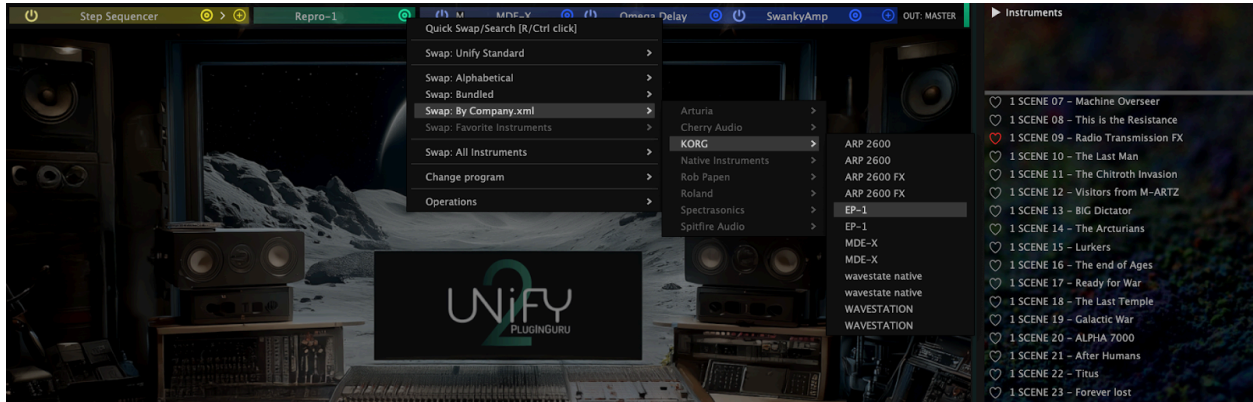
There are 3 types of 'plugin box' (with default colors yellow, green, and blue). The left most box on instrument layers is the 'MIDI plugins' box, which then feeds into the 'Instrument plugin' box, which sends audio into the 'Effect plugins' box.

The power button will appear on each MIDI and audio effect plugin, and enables disabling / bypassing that effect.

The 'Target' icon button opens up a menu to select/swap a new plugin or copy/paste the state.

You can drag (horizontally) and drop plugins within the effect boxes to re-arrange them (if more than one plugin is loaded for audio effects for example).

As mentioned in the Settings panel description, the Quick Swap options feature sub-sets that can be user defined.

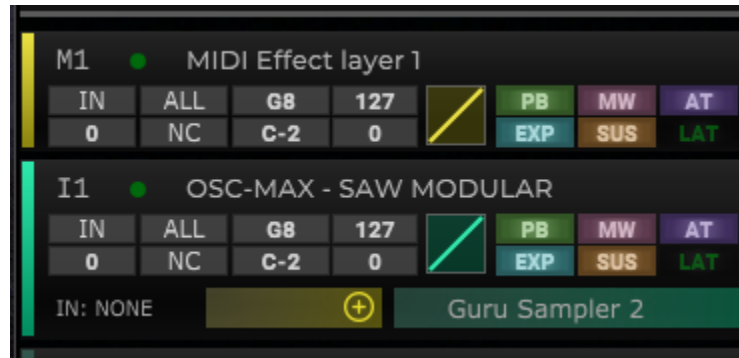


Right clicking on the Plugin boxes will open a Quick Search/Swap dialog.



MIDI layers only feature the 'MIDI effects' box, and Aux and Master layers only feature the 'Audio effects' box.

## MIDI and Instrument layer input controls



MIDI and Instrument layers both process MIDI data - and have multiple control buttons to select and filter MIDI data.

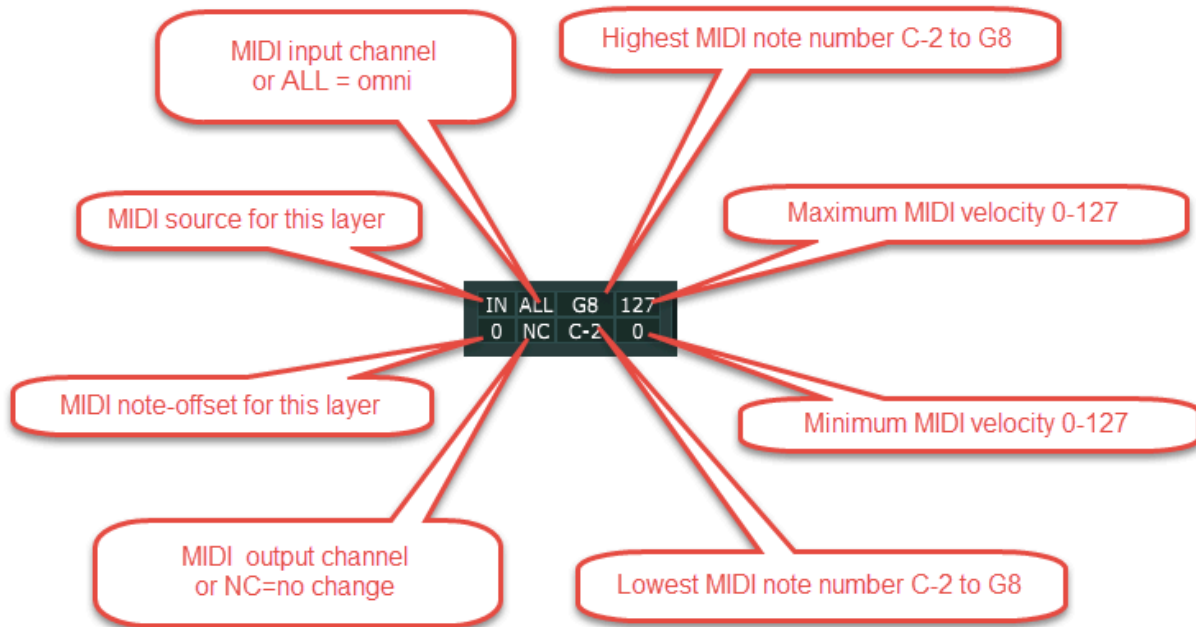
INST and MIDI layers can be *disabled* by clicking on the green MIDI indicator-dot at the left end of the layer. When a layer is disabled, it goes dark, and the flow of MIDI data to the layer is interrupted so it won't play any new notes (until you re-enable it), but any sustaining notes will continue to sound.

INST and MIDI layers can also be enabled/disabled using the top-level *midiEnable* and *midiToggle* parameters, which can be linked to macro knobs or used in [MIDI Controller Files](#).

See also [Parameter paths reference](#) for details, specifically the [Top-level paths](#) section.

The stream of MIDI data coming in to each MIDI or instrument layer is subject to three kinds of operations, all set using the cluster of MIDI-related controls at the left-hand side of the layer (see the diagram on the next page):

1. **Stream selection:** You can select whether the incoming MIDI comes directly from Unify's main MIDI input, or from the output of one of the patch's MIDI layers.
  - *MIDI Source* control
2. **Filtering:** You can choose to filter out MIDI note-events based on *pitch*, *velocity*, or both.
  - *MIDI input channel* and all four *MIDI note number* and *min/max velocity* controls
3. **Processing:** MIDI events can be altered before they are sent on to the plug-ins.
  - *MIDI output channel* and *MIDI note-offset* controls
  - *Velocity response graph* at the right



## MIDI cc filters and the Latch button



These buttons are by default in the 'ON' position except for the LAT (Latch) button.

The buttons are as follows:

PB = Pitch bend enable

MW = Mod wheel enable

AT = Aftertouch / Channel pressure enable

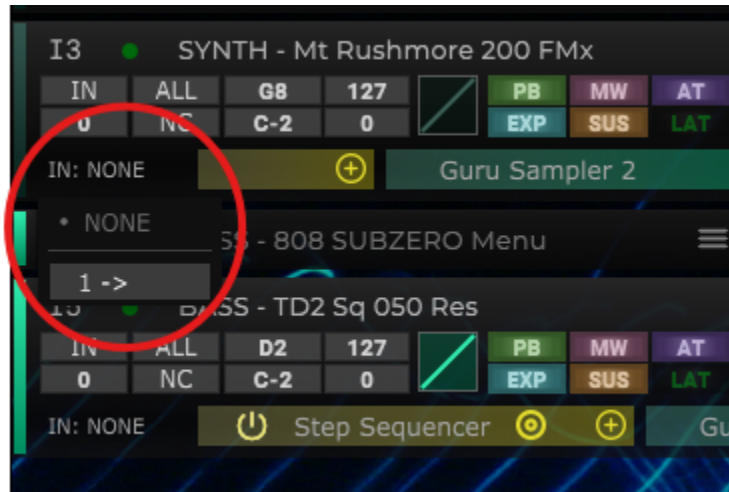
EXP = Expression pedal enable

SUS = Sustain pedal enable

These are always accessible MIDI filter shortcut buttons - they allow you to disable or enable pitch bend, mod wheel, aftertouch, expression pedal, sustain per layer - this can be really handy for example if you want pads on one layer and to play a lead in the other layer - so you could for example disable pitch bend on the pads layer and disable sustain on the lead layer.

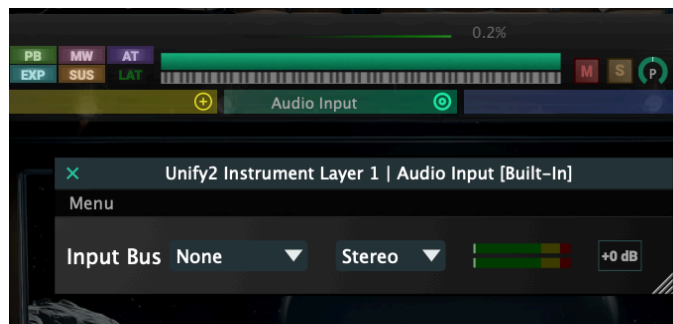
The last button in that area is the latch button, which becomes enabled if the MIDI source/input is set to Latch mode.

When you enable audio inputs or multiple audio outputs (either via the DAW settings or hardware interface settings in standalone mode) you will see a selector on the left and right hand side of instrument and aux layers.



These let you pick a hardware input to receive audio from or output to send audio to.

When starting with an empty/default layer, the 'sine wave synth' is loaded by default. In this case, selecting an audio input will replace that with the audio input plugin.

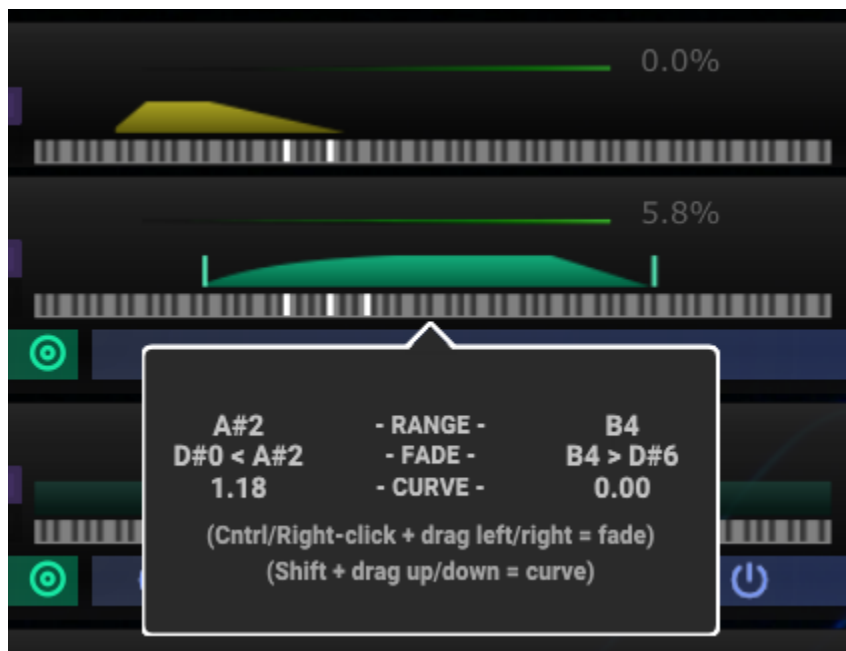


You can also replace the 'Audio Input' plugin with any instrument plugin that enables a side chain audio input such as a synth with a vocoder or compressor built in.

## MIDI and Instrument layers - zones editor/live keyboard view

The MIDI and instrument layers show a zones editor view to the right of the filter buttons - always visible in expanded layer mode. This editor lets you drag the start and end note zone for each layer as well as set velocity fade and curve amounts. The velocity fade and curve will affect MIDI velocity data passed on to the MIDI and instrument plugins.

The keyboard shows a live representation of the incoming notes and these are only shown as active if they are within the defined zone.



Note that the keyboard can be hidden if you prefer not to see it - either via the mini keyboard button at the right hand side of the layer controls, or as a permanent option in the UI settings panel.

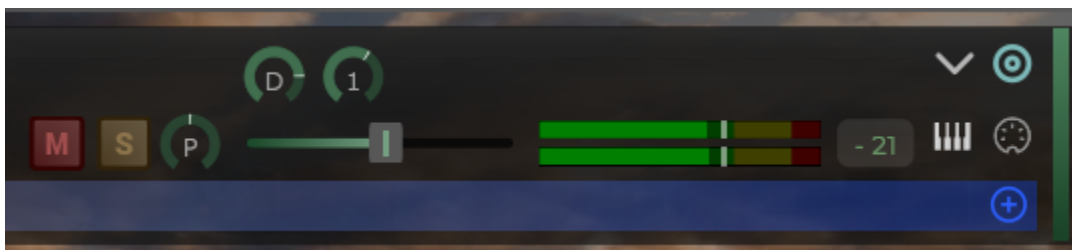
Above the keyboard view, there is a live 'CPU load' indicator value and moving graph showing you the current processing load of the layer. A 'CPU Load' indicator is also shown on the Aux and Master bus layers.

## Layer audio controls

The Layer audio controls are similar for instrument, aux and master layers. Mute, Solo, Pan and level controls are on the left.



When in 'expanded' view the instrument layers also show sends for the amount of direct to master send, and then a aux send control knob for each aux layer that has been added.

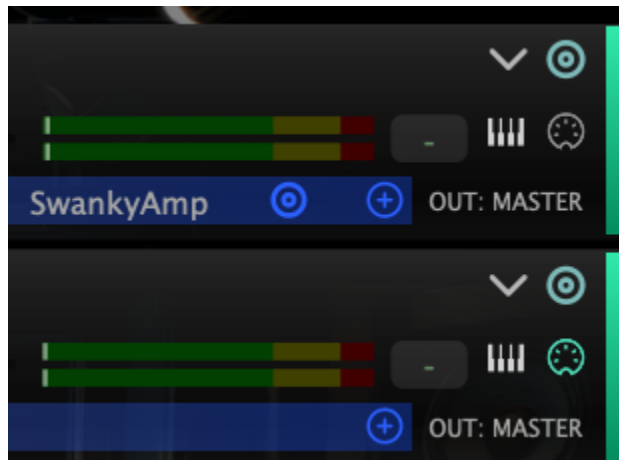


Option/Alt click whilst dragging the volume slider will make all sliders on all layers move at the same time - this is really handy for adjusting the total level being sent to aux and master layers.

The audio level indicator shows the instantaneous level, but also shows a gradually decreasing peak level indicator which fades out over a few seconds - this is helpful to determine if levels are getting too 'hot' and could potentially cause distortion when feeding into aux layers or the master bus - or to your sound-card/audio tracks in your DAW.

Generally you want to keep the audio peaking, at most, in the top of the yellow area for the loudest parts/layers.

## MIDI out



The MIDI out button is located immediately to the right of the mini keyboard toggle button.

You can enable MIDI out on MIDI and instrument layers - the MIDI icon can be toggled on or off.. it's used to send MIDI to the main MIDI hardware output (in standalone mode), or you can forward MIDI data from Unify to your other DAW tracks.

## Layer drag-and-drop

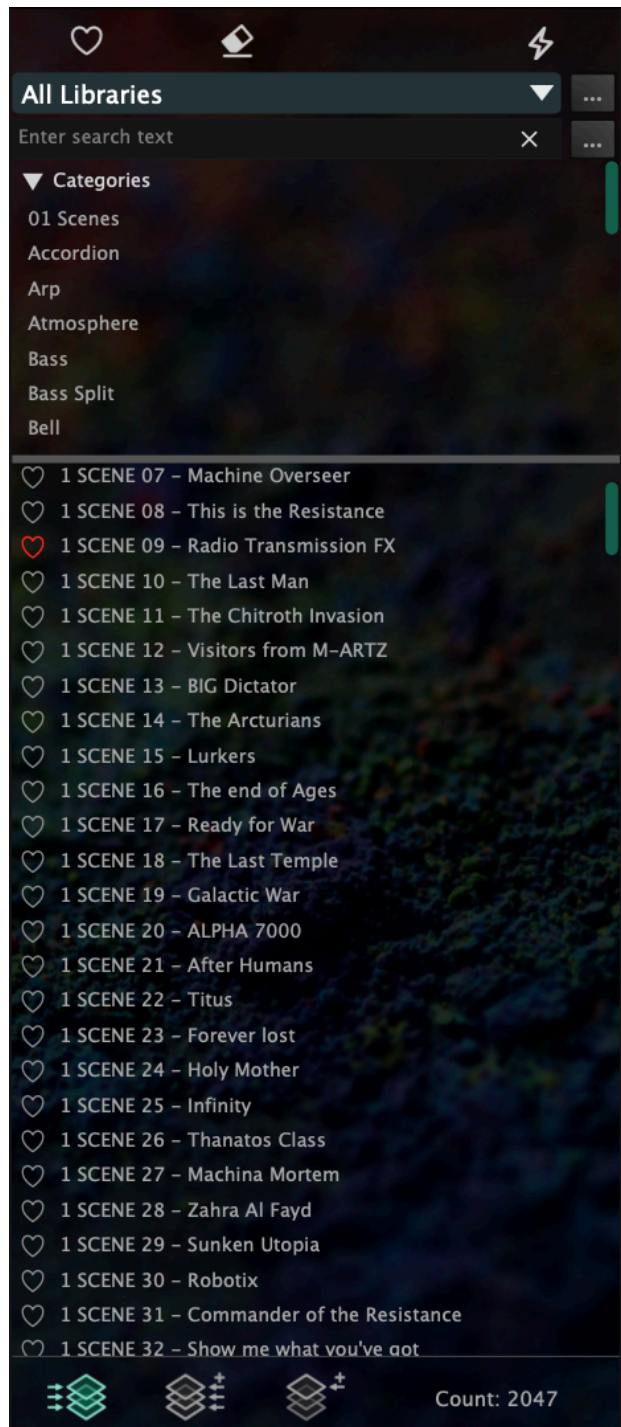
- “Within each group” means you can, e.g., drag the INST2 layer and drop it on top of the INST1 layer to reverse their order, but you can't drag an INST layer up among the MIDI layers. Each kind of layer always remains together with others of its kind.
- Drag using the **left mouse button** to click the layer's *title strip*.
- As you drag one layer over another, red lines will appear, to indicate where the dragged layer would go (either above the target layer, or below it) if you were to release the mouse button at that point.
- Unify will automatically update the layer numbering and all macro parameters linked to the layers.



## Patch browser

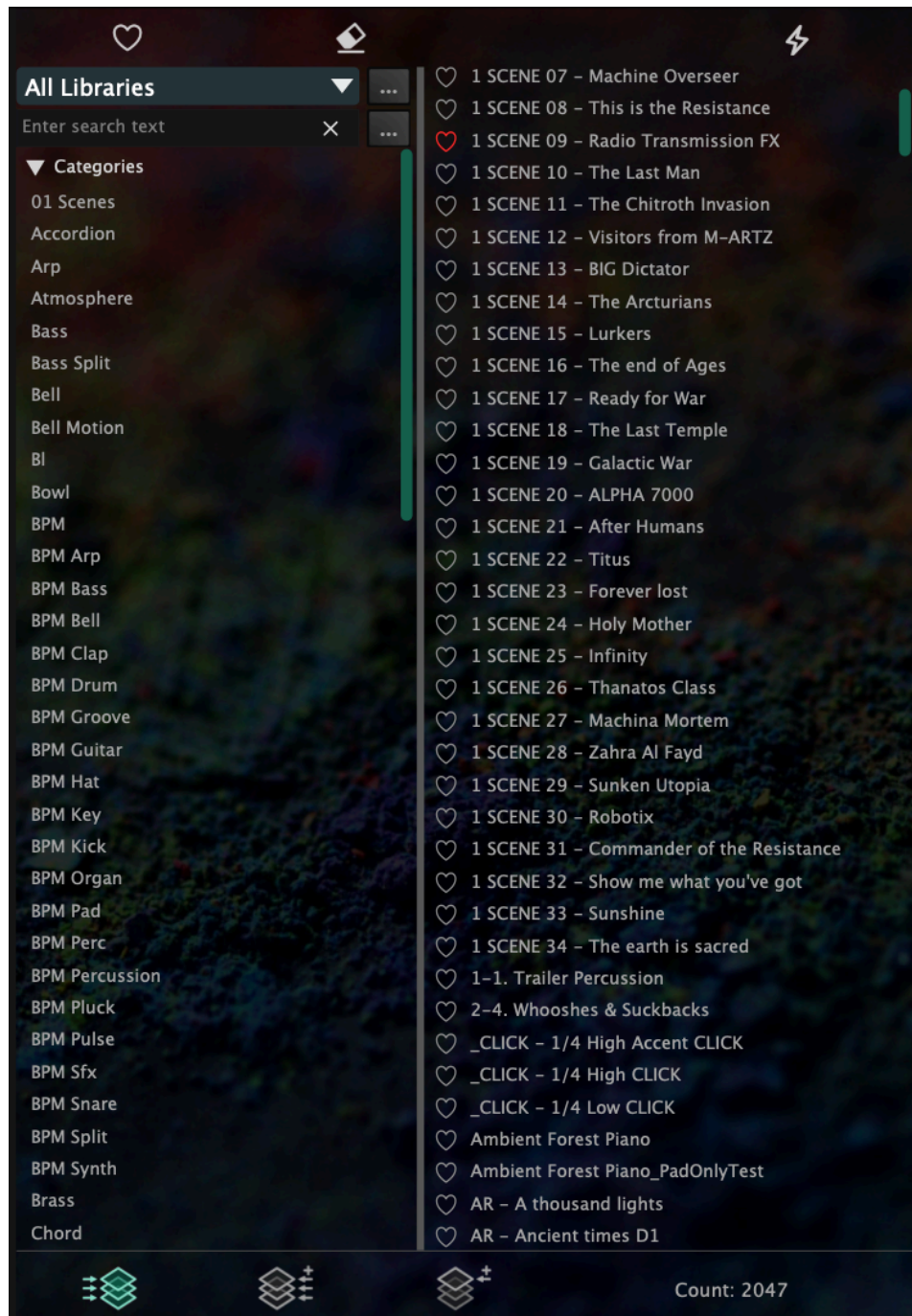
This is the default, single column patch browser view.

When hovering the mouse/trackpad pointer over a patch name, the top/search area shows the meta-data for that patch.



## Side by side mode

The patch browser is, by default, displayed as a single column. There are Library Filtering and Patch Name Search options at the top. If you have a wide-screen, and stretch the window wide enough, the Patch browser will appear as two columns. This mode can make it easier to see the list of patches and tags in a Unify library(ies).

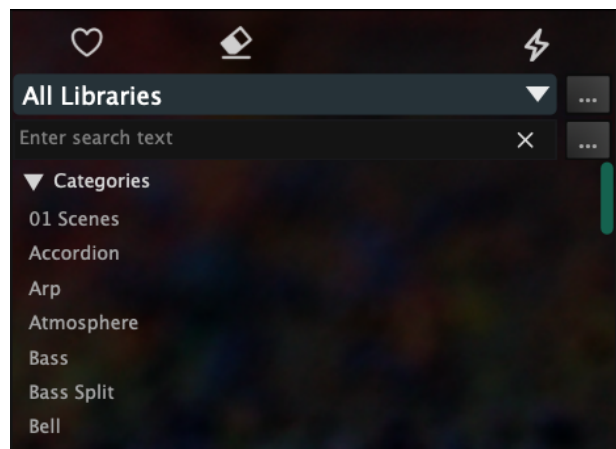


## Searching

The top of the Patch browser interface features 3 buttons:

1. The Heart button filters the list by your favorites
2. The Erase button clears the current filtering settings
3. The Bolt button triggers a library re-scan

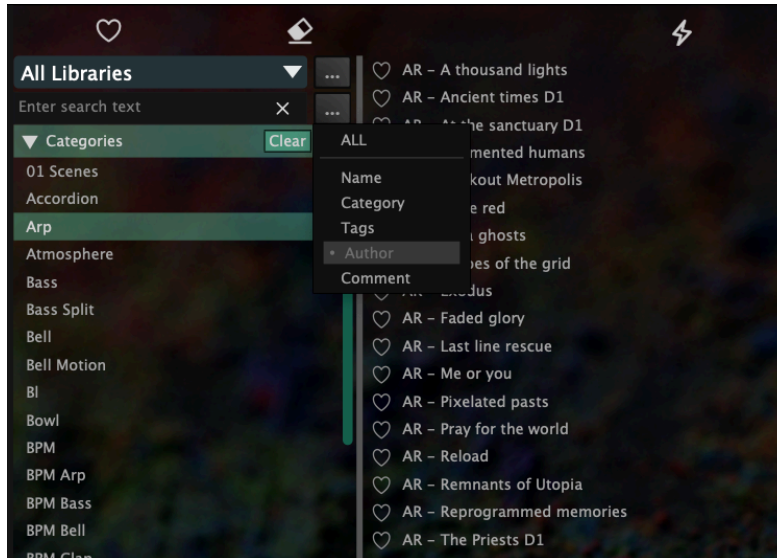
Option 3 finds newly added patches to your Unify environment. Holding the alt/option key while clicking the Bolt button triggers a full patch database rebuild, which can take some time.



The Library pull-down list, lets you select a Library from a list of installed Unify libraries.

Type text into the “Enter search text” box to find presets with the nominated search string. By default, the search applies to all patches in all libraries installed. Clicking the ‘...’ button immediately to the right of the “Enter search text” box restricts a search to a specific patch metadata category.

Clicking the “x” in the “Enter search text” box clears any existing text in the box.



Clicking the disclosure triangle next to Categories, Tags, Instruments or Authors reveals a list of searchable metadata items that can be used to filter the patch list. Use 'cntrl' and left-click to select multiple entries.

## Patch loading modes and right click options

There are 3 main loading modes. Selecting a specific mode is achieved by clicking the icons at the bottom of the Patch browser panel:



### *Normal mode / Replace all layers:*

The left-most icon: When you click on a patch in the patch list, all currently loaded layers and embedded instances are removed and replaced with those of the new selected patch.

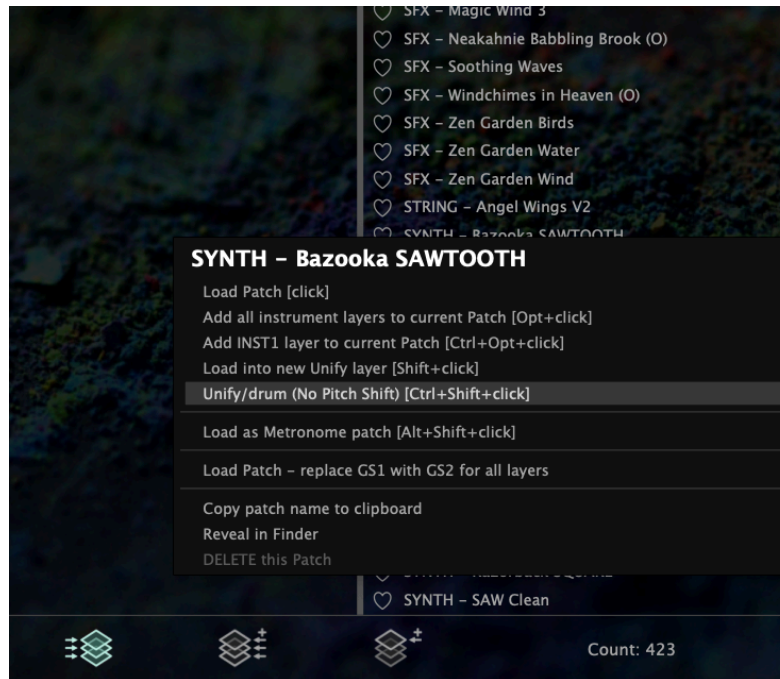
### *Insert new Unify:*

All layers of the selected patch will be loaded into a new Unify instance in a new instrument layer in the currently loaded patch.

### *Load layer 1 only:*

Only the first instrument layer of the selected patch will be loaded into the current patch as a new Unify layer.

Right-clicking on a patch name displays a menu of loading modes and other options pops up:



The options here are:

- Load Patch [click] - loads the selected patch
- Add all instrument layers to current Patch [Opt+Click] - All layers are added as new layers to the current patch
- Add INST1 layer to current Patch [Ctrl+Opt+Click] - Adds the first Instrument layer in the selected patch to the current patch
- Load into new Unify layer [Shift+Click] - loads the selected patch as a new Unify layer into the current patch
- Unify/Drum (No pitch shift) [Ctrl+Shift+Click] - Patches are loaded where Guru Sampler / Guru Sampler 2 has no pitch shifting.
- Load as Metronome Patch [Alt/Opt+Shift+Click] - The selected patch is loaded as a 'Metronome Unify' instance; the Metronome must be enabled to hear it.
- Load Patch - replace GS1 with GS2 for all layers - all GuruSampler instruments will be replaced with an equivalent GuruSampler 2 instrument. The patch is converted automatically with an approximation of the original modulation settings.

The DELETE this Patch option is only available for user patches.

## Transport & controls panel

The bottom section of the UI features an always visible, Transport control panel with frequently used controls and transport status information.



At the very left of this panel are the following five (5) buttons:



The button functions from left to right are:

1. MIDI Panic - turns off all notes / stops currently sounding instruments.
2. Latched notes off - stops all notes currently latched by the Unify layer latch feature.
3. Virtual keyboard toggle - open the virtual keyboard panel.
4. Macro controls toggle - open the macro knobs control panel.
5. Plugin Windows to Front button - brings all hosted plugin windows to front (not supported by all DAWs).

In the center of the panel is the transport control interface with the following items:



The controls from left to right are:

- *MIDI trigger note range* - the range of notes that will affect that transport trigger
  - Click and drag to change the top and bottom note, or right click to set the note box into learning mode - then play a key to set the note.
  - This range is saved per patch if it has been set to something other than the default/full range.
- *Trigger mode:*
  - **MAN** - Manual / Start/Stop - click the Play/Stop to Start/Stop the transport.
  - **M.QS** - MIDI / Quick Stop - play/stop happens when notes are held or released.
  - **M.RS** - MIDI / Restart - playing notes starts the transport, playing again resets the position.
  - **M.MS** - MIDI / Manual Stop - playing notes starts the transport, click Stop to stop the transport
  - **M.LA** - MIDI / Latch - Transport toggles on/off in support of using Latch modes.

- **M.1B, M.2B ... M.16B** - MIDI 1, 2 ... 16 bars - playing notes starts the transport, on release, the transport keeps running for the defined number of bars. This can be useful to keep a drum groove, metronome backing, arpeggiation running whilst releasing your hands from the keyboard.
- *Host Follow Mode* (only available within a host/DAW)
  - No host-follow - free run of the transport regardless of host mode, is useful for auditioning patches and using Unify at tempos other than the DAW Tempo.
  - Host-follow when running - the transport will run locally free from the host, until the host transport starts, then it will lock to the host transport. (This is the default and probably most useful mode.)
  - Host follow always. Only run the transport when the host transport is running
- *Record button*: Start recording - the chevron button above opens the recording options panel. Right click the record button to open the most recently recorded files.
- *Play/Stop button* - for manual start and stop of the transport and for stopping a recording.
- *Rewind* - brings the transport back to the beginning of musical time / 0 seconds elapsed.
- *Elapsed time in seconds* (indicator only).
- *Elapsed time in bars and beats* (indicator only).
- *Tempo indicator and control* - you can click and drag to set a new tempo or right click to open a pop-up to directly type in the tempo value.
- *Metronome toggle* - turns the metronome on or off. Right-click opens the Metronome Unify instance. The chevron button opens the Metronome Options panel.
- *Time signature* - Click and drag to change the values (these values are locked to the host in plugin mode).
- *Ref bpm* - the bpm value of the original patch that was loaded.

To the right of the transport area are seven (7) buttons:



The function of these buttons from left to right is:

- *Background image brightness control* - to reduce the brightness of the current image - it can be useful if you find it distracting.
- *Play mode toggle* - enter/exit the play mode view.
- *Save (Floppy Disk icon)* - left click to open the save patch dialog. Hold the ALT/Option key to update (quick save) the current patch (only if not a protected patch). The save icon will change to orange if the ALT/Option key is held whilst mousing over, and will flash green on a successful update or red if saving was blocked.
- *Revert/Undo button* - Opens a list of the last state saves (see the UI options).

- *Init button* - Resets Unify to the default start up patch, clearing all layers. Opt+Click resets the patch but retains the INST 1 Layer.
- *Expand or Contract* - this toggle button will set all layers to either expanded or contracted mode - the icon changes accordingly.
- *Toggle browser button* - Show or hide the Patch browser - the icon is highlighted when the browser is open.

## Virtual keyboard / MIDI controls



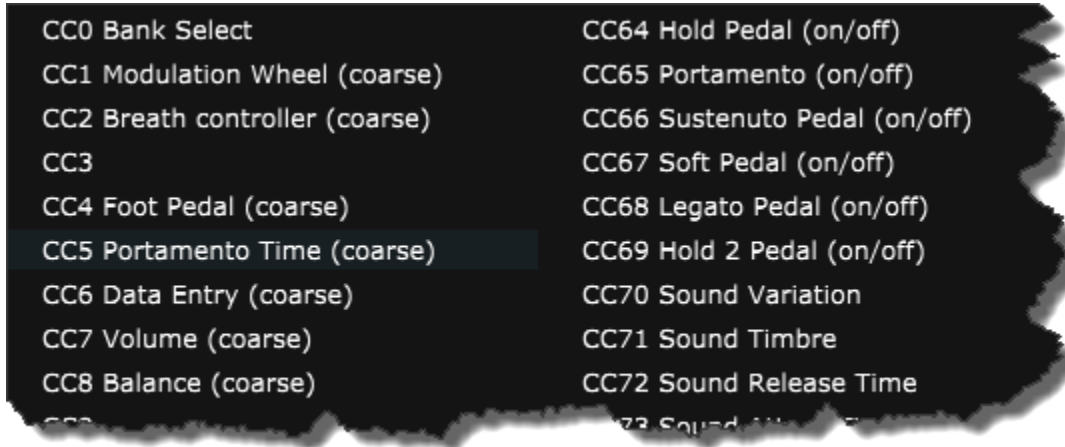
This panel shows a virtual keyboard with MIDI CC control sliders on the left hand side.

Active notes will be shown in different colors as you play from an external controller.

You can play this keyboard by clicking on the keys.

The first (unnumbered) slider is a bi-direction pitch bender, after that a modwheel/CC1 control and 3 more assignable CC controllers.

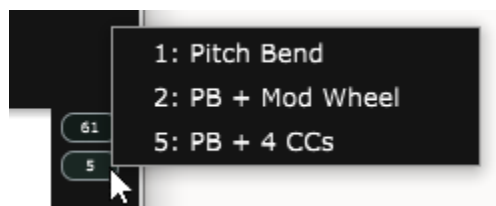
Right-click on a CC control slider to select a different CC number:



To the far right hand side of the keyboard are four (4) buttons:

The *A button* selects the number of keys based on various popular MIDI controller keyboard sizes; 25, 32, 37 ... 88. 'Auto', the default value, adjusts the number of keys shown relative to the Unify window size.

The *S button* picks how many control sliders are available on the left; there are 3 options:



The *NC button* converts the output internal MIDI channel; 1 through 16.

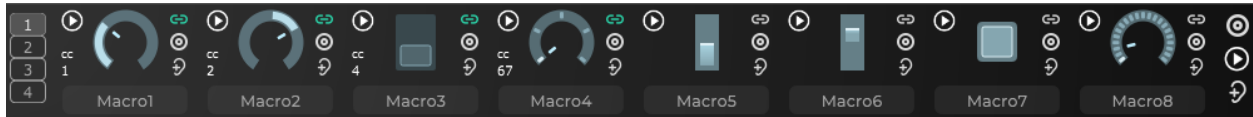


The global MIDI Channel Conversion setting is NOT saved with the patch.

New instances of Unify will always come up with this control set to the default "NC" (No Change), so Unify works normally. This function is solely intended for testing; it is not a substitute for changing the MIDI-output channel assignment of your MIDI controller(s).

The *last button* (0) is a global transpose control value for the virtual keys; value is 0 by default.

## Macro controls panel



The Macro Controls panel shows 8 controls at a time.

In Unify 2 we have multiple control types as shown in the screenshot above.

Click the small circle to the right of each knob to see the menu where these knob types can be chosen towards the bottom of the menu under the “Control Type” entry.

The image above (from Left to Right) shows the following knob types:

- Regular knob control
- Bi-directional knob control
- Toggle switch
- Selector (with variable number of detents)
- Fader
- Bi-directional fader
- Momentary switch
- Another selector is shown with more detents.

The macro controls also show the currently selected value in an area around each knob and the modulated/externally controlled value.

Right clicking the control will open the parameter links panel, or you can get to that via the target/operations “small circle” button to the Right of each knob.

More information on macro links and the Macro controls panel can be found in the following section: [Unify2UserGuide](#)

## Play mode view



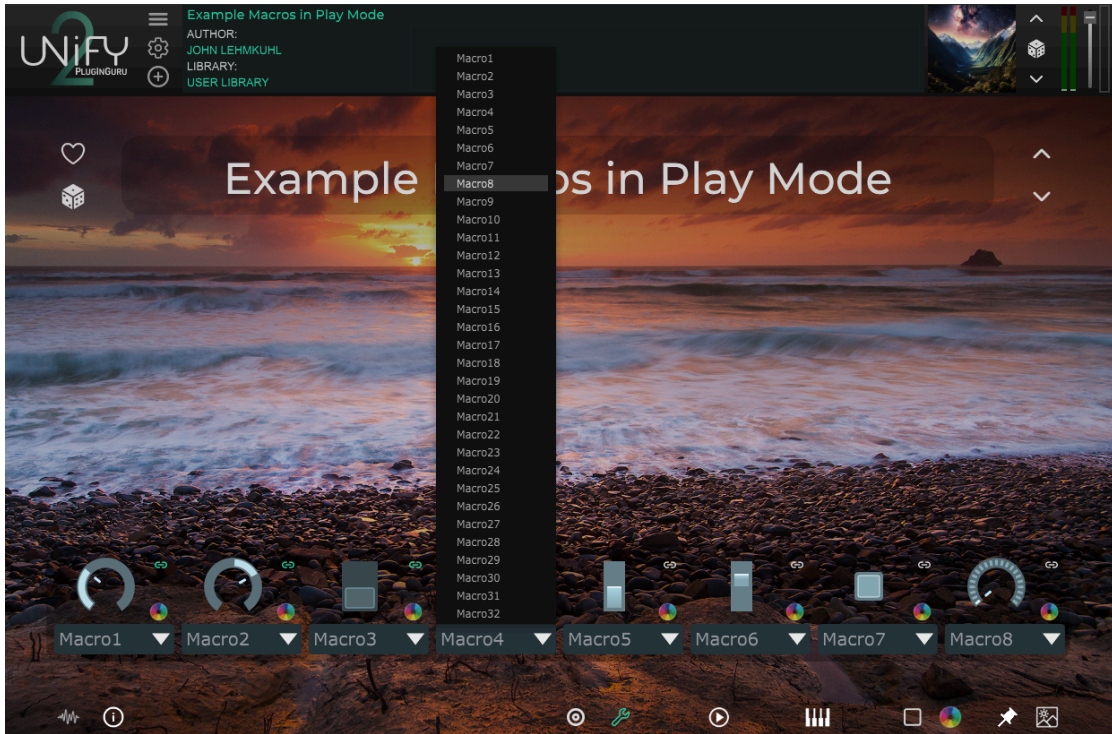
This view is intended to provide a minimal interface for just playing patches without being concerned about the complexities of the patch design and layers that often create complex sounds.

*Play mode* is accessed by clicking the 'mountain sunset' picture icon; immediately to the left of the Save (Disk) icon.

Play Mode accesses the same Macro Controls that are displayed in the main layers view Macro Controls panel. The default number of Macro controls shown is set to 8.

The number of visible Macro controls can be changed by clicking the Spanner button. After clicking the Spanner button a new Target icon appears immediately to the left of the highlighted Spanner icon. Click the Target icon to change the number of visible Macro controls.

Visible knobs can be assigned to a different Macro Control by clicking the Macro name:



The current setting of available controls and selected macros is saved with the Patch data. For example, here we changed the layout and order of assigned controls.



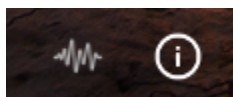
The Play mode panel lets you pick a different Background image for your workspace/patch. Background images are saved with your patch.

To change an image, click the *Picture button* immediately to the right of the Pin icon. Click an image to select it.



Other Play mode buttons:

At the far left hand side:



1. The *Wave icon* turns the Waveform visualizer on/off
2. The *Information icon* is an Image credit button.



To the right of the Spanner icon (Tool button):



The buttons from left to right are:

1. *Macro LFOs* play/stop master control
2. *Keyboard toggle* that turns the virtual MIDI keyboard on/off
3. *Checkbox* that enables the background color filter.
4. *Color Picker* that opens the Color Picker panel to overlay on top of the image. This can be a solid color if you want to hide images completely, or you can set some opacity to allow the background image to filter through. To keep this color for future sessions, click on 'set as default' inside the colour picker.
5. *Pin button* pins the current image; this currently only works on factory built-in images (available via the Image Selection panel).
6. The *Image Select button* opens the Images Selection panel.

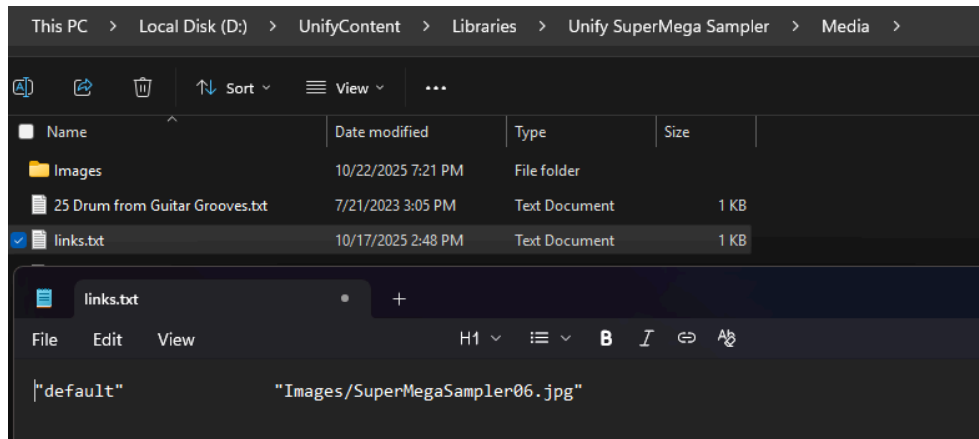
Further notes about image selection:

When an image is 'pinned' the *Pin button* is vertical. This means every time you load a patch, the image selected will be the *Pinned Image*.

If the *Pin button* is at an angle, when you load a patch, if there is either a built-in (embedded) image or a link file image associated with the patch, the background image will be replaced with the saved patch image. This image is also the same image that appears in the Layers view.

If you would like to set a default image for your entire user library or other library, you can edit the links file and set a 'default' entry and point that entry to an image in a sub-folder of the library.

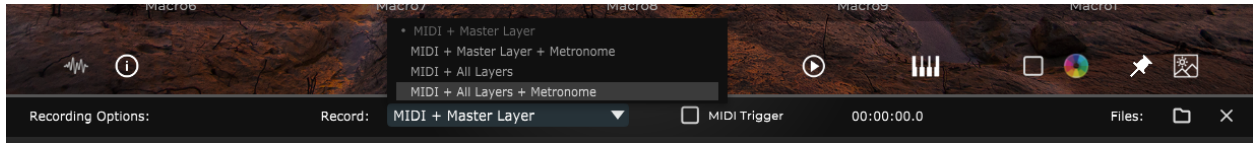
For example (as has been done for the SuperMegaSampler library):



You can also set images per patch by adding entries with the patch name and a specific path, each entry on its own line in the links file.

## Recording controls panel

Open the *Recording Options* panel by clicking the upward facing chevron button above the *Record button*; the Red Circle icon:



This Recording Options panel determines which layers of the main Unify instance will be recorded.

There are four (4) recording options:

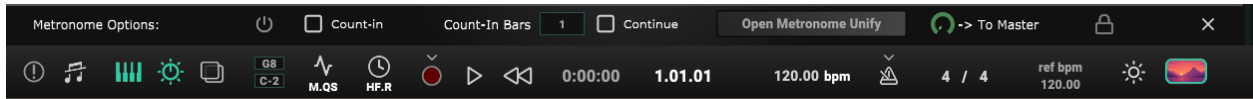
1. MIDI + Master Layer
2. MIDI + Master Layer + Metronome
3. MIDI + All Layers
4. MIDI + All Layers + Metronome

Recording of stems for ALL Layers is a handy way to export your live playing for later mix-down in your DAW.

There is a MIDI Trigger check box to the right of the MIDI recording options drop-down list that allows the Record button to be clicked, with recording only starting at the first MIDI note event received.

The Recording Options panel also shows the current recording time. The *Files button* at the far right opens your computer's Files browser to the last recording session files location. If there are no recorded sessions, the Files button does nothing.

## Metronome options panel and the metronome



The *Metronome Options Panel* is opened by clicking on the upward chevron icon above the Metronome button.

The controls from left to right are as follows:

1. *Power/On-off* button.
2. *Count-in* checkbox - enable metronome count in (the metronome will 'click' during the count in bars period).
3. *Count-in bars* value box. Set the number of bars for the count-in period.
4. *Continue* checkbox keeps the metronome 'ticking' after the count-in period is finished.
5. *The Open Metronome Unify* button opens the metronome instance. The same can be achieved by right-clicking the *Metronome* icon
6. The *To master* knob mixes the audio from the Metronome instance with the Master layer of the top Unify instance.
7. The *Lock On/off* button locks the Metronome layer patch when changing patches in the top Unify instance.

Further information about the metronome instance and metronome grooves capability is in the following section: [Unify2UserGuide](#)

## Creating / Editing new sounds

# Creating your first patch (Tutorial)

Lets work through a simple example of creating and saving a basic piano-and-synth patch. This tutorial assumes no prior Unify knowledge and takes a step-by-step, *tutorial* approach.

You'll be working almost entirely in Unify's Layers view.

You needn't click that link just yet; this page is meant as a beginner's tutorial, so we'll guide you through every step, without getting into too much detail.

## Start from an INIT patch

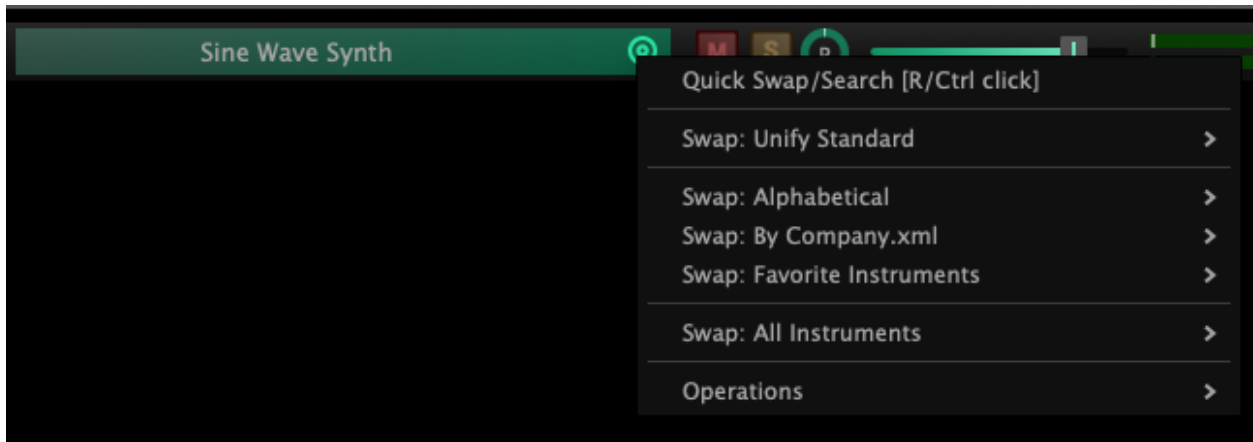
To create a new patch “from scratch”, click the *Initialize Patch* icon (atomic symbol) button in the Toolbar and Controls Panel; bottom of the Unify window, third icon from the right. A new preset is created, with a single Instrument layer with the Sine Wave Synth instrument loaded. An empty Master layer is also created.



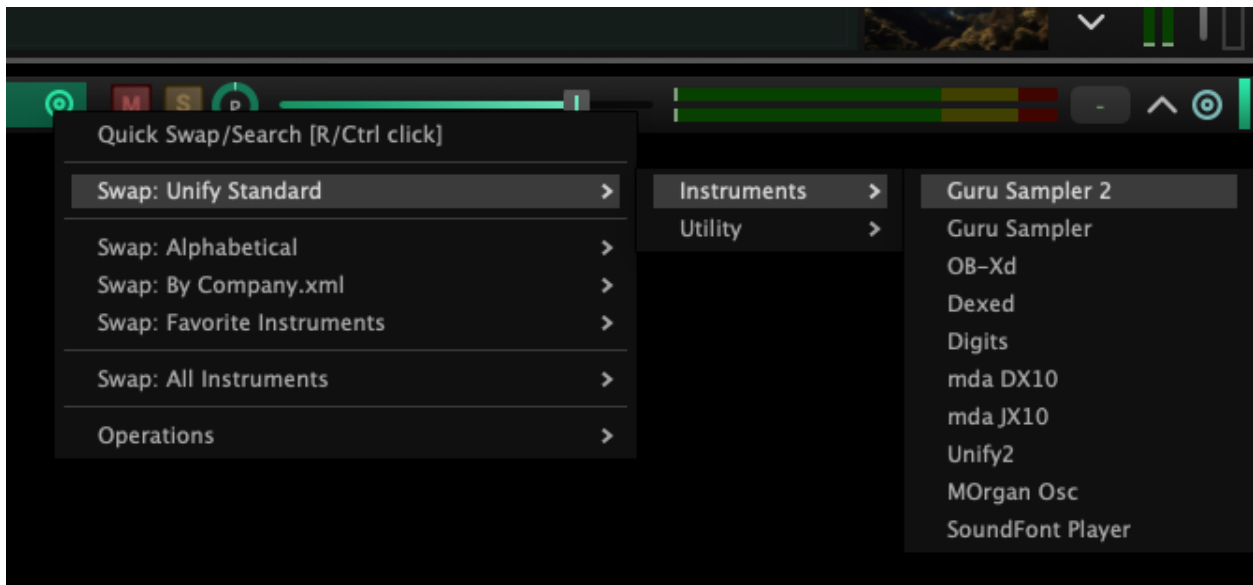
Play a few notes on your MIDI keyboard to make certain Unify is working correctly. If you don't hear any sound, go back and check that you have correctly set up your MIDI and Audio settings.

## Create the piano layer

Click the *Operations button* (two concentric circles icon) to the right of the words *Sine Wave Synth* to open the *Instrument Operations* menu:



Roll the mouse pointer to the right, over the *Swap: Unify Standard* item, until the second-level menu opens (“Instruments” and “Special”). Keep rolling across *Instruments* until the third-level menu opens, and click the first item *Guru Sampler 2*:



The “Sine Wave Synth” instrument is replaced with the “Guru Sampler 2” instrument, and the Guru Sampler 2 window is displayed.

Don't worry about the details at this point; just look for the row of three pop-up menus to the right of the “Sample Selection” area. Click the middle drop-down list and select “Keyboard”. From the next drop-down list select “HR Pop Piano f”, as shown below:

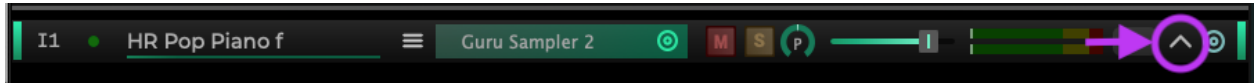


Play a few notes on your MIDI keyboard, to ensure that you are hearing piano tones, before moving on.

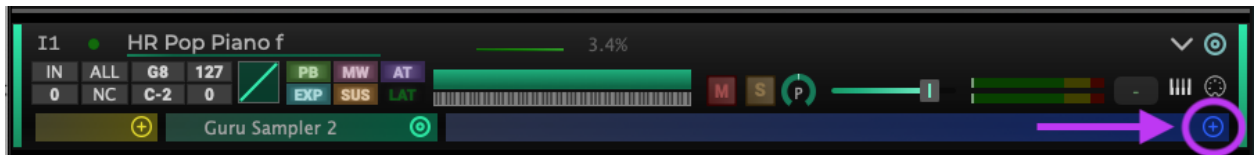
Close the *Guru Sampler 2* window by clicking on the “x” at the top-left of the window.

## Add an audio effect

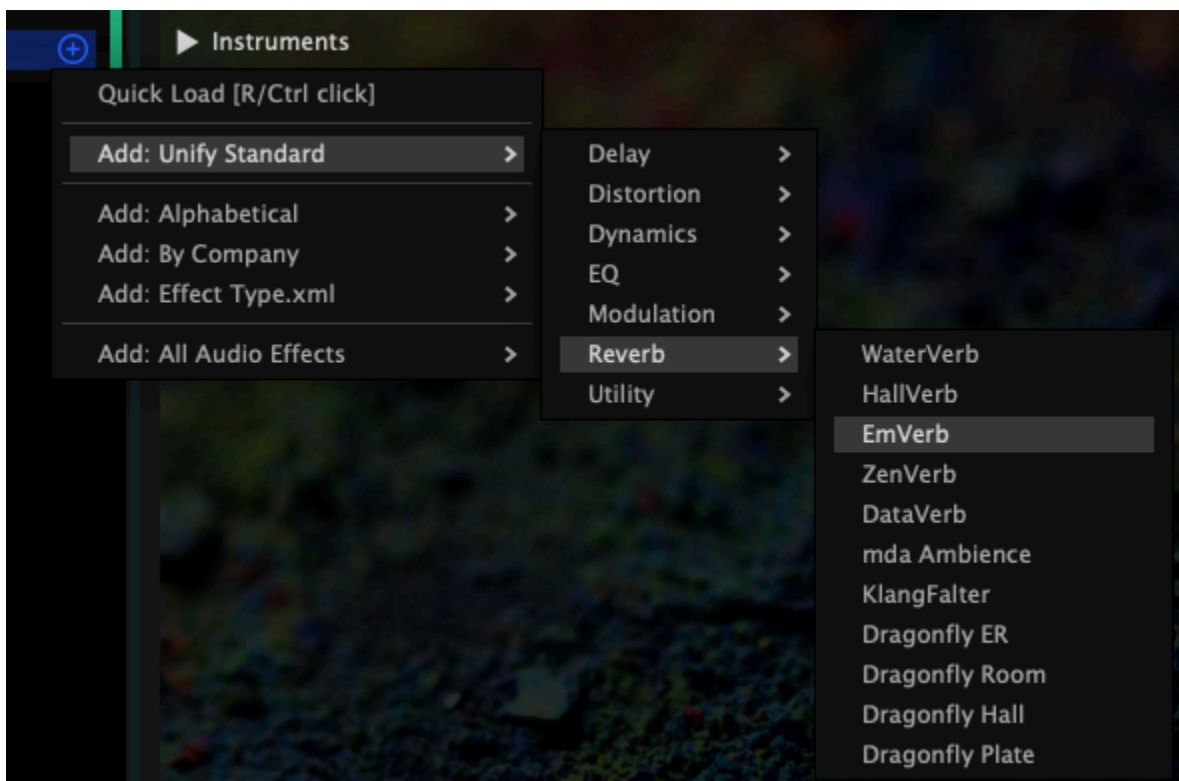
Expand the *Instrument* layer by clicking the chevron as shown in the image below.



In the blue *Audio Effects* box click on the *Add button* (+ sign in a circle) as shown below:



From the displayed menu select “Unify Standard” through “Reverb” to “EmVerb” and click on that:



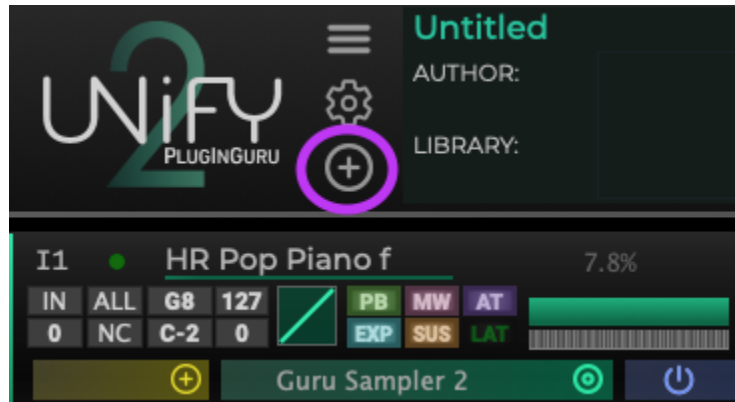
You have just added an instance of the *EmVerb* plug-in to the output of *Guru Sampler 2*. In the *EmVerb* GUI which then appears, adjust the knobs approximately as follows:



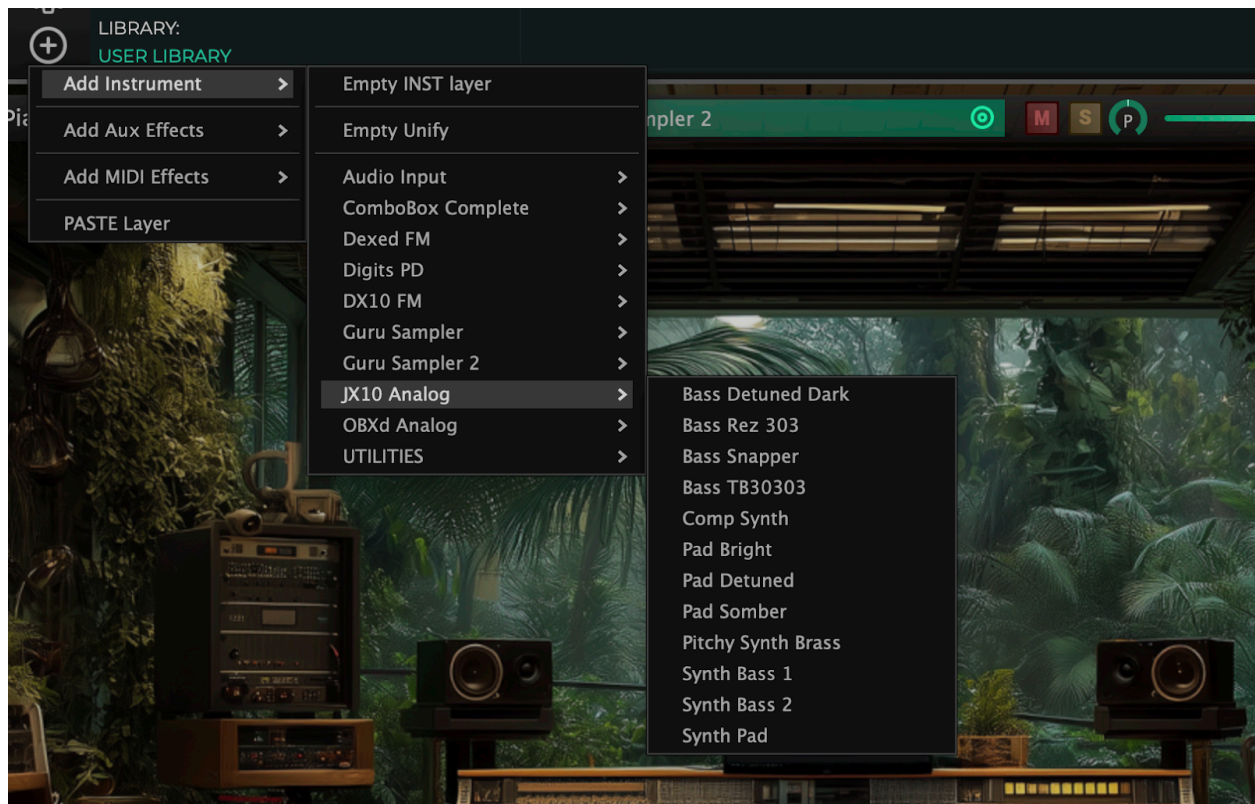
Close the *EmVerb* GUI window by clicking on the “x” at the top-left of the window.

## Add the synth layer

Locate the *Add Layer button* to the right of the Unify 2 logo, at the top-left corner of the Unify UI.

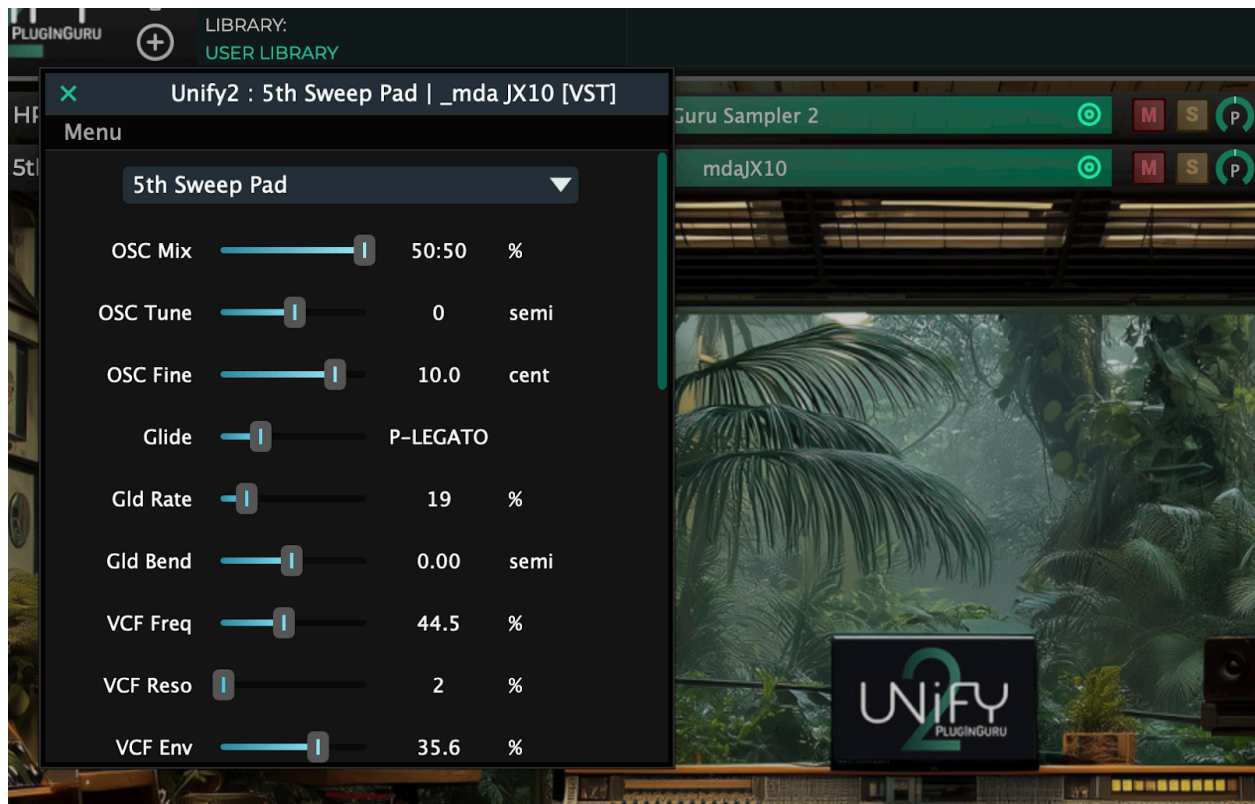


Click it to open the *Add Layer menu*, which is another system of cascading sub-menus with various options. To add a new Instrument Layer, select “Add Instrument” > “JX10 Analog” > “Synth Pad” to add an instance of the *mdaJX10* synthesizer:



A second *Instrument layer* is added to the patch with the instrument name *mdaJX10*.

Click the *mdaJX10* instrument name in the second *Instrument layer* to open the instrument window:

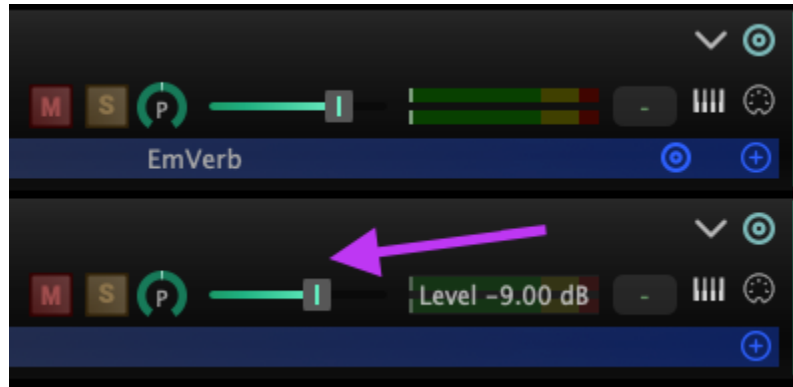


At this point, we don't need to make any changes, so just close it by clicking the “x” in the top left of the *mdaJX10* instrument window.

Play a few notes on your MIDI keyboard to confirm that you hear piano and synth tones layered together.

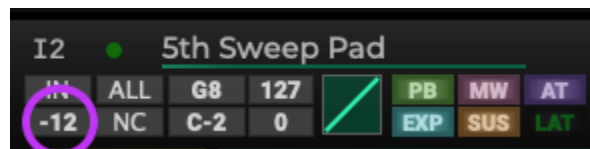
## Adjusting the layer balance and octaves

You'll probably want to make at least two further changes. First, the synth layer will be too loud, so reduce the **layer level fader** down to about -9 dB.



Second, the layer preset was created with a MIDI pitch offset of -12 semitones, so the synth layer sounds an octave lower than the piano layer. To change this, first click the *Chevron* (^) icon to expand the view of the **I2** instrument layer.

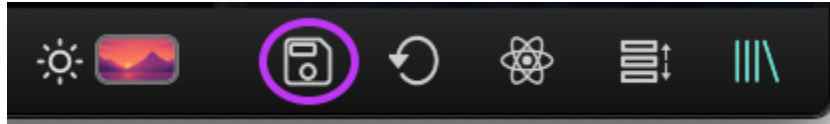
Locate the **MIDI note-offset control** at the bottom-left corner of the *MIDI-controls cluster* for the **I2** instrument layer, where it says “-12”:



Roll your mouse pointer over the control, press the mouse button, *drag upwards* (keeping the mouse button down) until the value reads “0”, then release. Play some notes to verify that the two layers are now sounding in the same octave.

## Save your new patch

At this point, you should **SAVE** your new patch. Locate and click the **Save button** in the *Transport and Controls* toolbar ( floppy disk icon) in the lower right-hand area of the Unify UI:



This will open the **Patch-save dialog**, which consists of several individual text-edit boxes.

Edit the items as shown below, then click the “Save As...” button.

Patch Name: KEY - Piano and Brass

Library Name: User Library

Categories: KEY

Tags: Brass, Synth, Keyboard, Bright

Authors: John Doe Replace with default author

Variations: Click to choose variations text file...

Key Zones: Click to choose key-zones text file...

Comment:

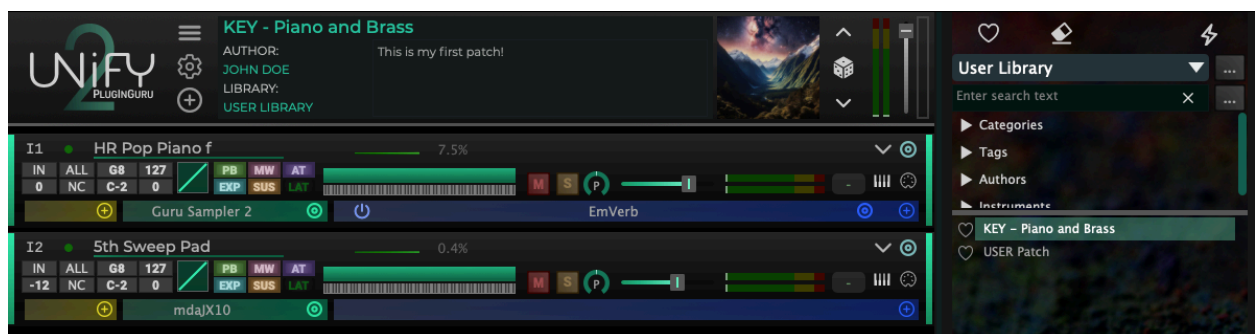
This is my first patch!

Lock Master Effects  Save metronome state

Cancel Update Save As... Save Layer Patches...

After you click the “Save As...” button, a standard file-save dialog opens, which will be different for Windows and Apple computers, but should be familiar to you. In this case, we have selected *User Library* as the library for this patch, but if you want, you can choose another one, or select “New Library...” at the bottom of the Library menu and enter a new name of your choice. If you choose the latter, Unify will then automatically create a folder with that name in Unify's main *Libraries* folder, and a sub-folder called “Patches” inside it, and save your new patch file there.

Click the “Save As...” button in the file-save dialog to confirm. The file-save dialog will close, and you should see the information you entered in the “patch metadata area” at the top of the Unify UI, and the name of your patch added (in alphabetical order) to the **Patch list** in the sidebar:



Congratulations. You just made your first patch in Unify!

# Macro controls, modulation and parameter links

***We will update this section of the User Manual with more information in due course.***

Macro parameter links in Unify 2 work the same way as in Unify 1. The main difference with the macro knobs in Unify 2 is that we now have several different types of Macro controllers, including knobs, faders, selectors, toggles and buttons.

Furthermore, each Macro now has a built in LFO for modulating Macro parameters. The LFO settings are in the parameter link editor panel. The 'remote' buttons for arming and start/stopping the Macro LFO controls LFOs allow to start and stop playback of those LFOs at any time, including control from a Macro; as such, you can use an external automation to turn the LFOs on and off.



Clicking the Chain Link icon at the top-right of each Macro knob in the Macro view of the main Unify UI, allows that knob to respond to, and transmit, a MIDI CC# control value. This is a globally set and doesn't change from patch to patch (if you're in a live setting, you want the controls on your keyboard controller to stay the same from patch to patch.)

However, if you load a patch as an embedded Unify Layer\*, THAT Layer's macro knobs can be set to different MIDI CC#'s and will be saved/recalled with those settings since it is inside an embedded Unify Layer. This gives you flexibility if you need to have different knobs do different things from song to song.

\*Any layer can be converted into a "Unify layer" from the far right target button menu with the "MOVE into new embedded Unify" option.

New to Unify 2, each Macro Knob now has a built-in modulation generator that you can think of like an LFO except this is far more flexible and powerful because there is no limit to how many parameters you assign to this knob and each parameter can have a different value path. An assignment to the cutoff filter could be a simple curve while you have the filter resonance changing doing entirely different things over the motion of the Macro Knob.

Please refer to the Unify 1 user guide for the moment for extra information on parameter links/macro controls:

<https://pluginguru.net/unify/manual/doku.php?id=realtime-params>

## Controlling Unify using DAW automation

Most plug-ins expose at least some of their parameters to the host application (e.g. DAW), so that those parameter values can be changed smoothly over time using a mechanism called *automation*. Parameters exposed for automation are said to be *automatable*.

For each automatable parameter, the host application needs to know (at least) the parameter's *name* and its *minimum and maximum values*. Certain plug-in formats (e.g. VST, VST3, AU) also specify more details, such as whether a parameter is *linear* or *logarithmic*, in its *unit of measure*, and so on.

### Unify exposes its Macro parameters for automation

The Unify plug-in exposes 32 automatable parameters, which correspond to each of the macro knobs. The host only knows that there are parameters, named “Macro1” through “Macro32”, and that their values may range (linearly) from 0 to 1 (i.e., 100%). The precise interpretation of parameter values is done inside Unify, using the macro parameters system.

This is very nearly all a User needs to know about Unify and automation, other than the following details:

1. Automation changes to unused Unify macros (i.e., macros to which no actual parameters are linked) are ignored.
2. Even if you rename a Macro knob in Unify, your DAW will still only show “Macro1”, “Macro2”, etc.
3. In most DAWs, you can manually adjust macro knobs during recording, and the DAW will record the sequence of changes and be able to replay it (as a sequence of automation messages) later.
4. You must be careful about using both MIDI CC assignments and DAW automation as discussed further below.

### Potential problems using both MIDI CC assignments and Automation

DAWs can record MIDI CC activity as well as automation. If you link any of your macro knobs to MIDI CCs, and manipulate those MIDI CC controls *while recording automation*, your DAW will record *both* the MIDI CC activity and the automation changes, and will attempt to send both into Unify on playback. This will very likely result in poor playback quality, as the two control mechanisms “fight” for dominance.

We recommend that you only use MIDI CC recording in your DAW for changes related to real-time musical expression; in other words, do NOT record DAW automation at the same time. Use DAW automation only for programmed changes, such as fade-ins, fade-outs and other transitions.

# Advanced usage of the Metronome Unify

The Metronome is a new function in Unify 2 that adds a second Unify plug-in instance to each patch. This special Unify Metronome instance is designed to trigger whatever sounds you wish to use as your metronome for jamming and recording. When you think of a Metronome you think of a static click that you can practice your Hannon piano exercises to... but thanks to this “side car” Unify, you have a complete playground for designing a groove that is triggered when the clock starts as long as the Metronome is turned ON.

## How to turn the Metronome ON

There are two ways to turn on the Metronome:

1. Click the Chevron above the Metronome Icon. Press the Power button in the displayed *Metronome Options* toolbar; or
2. Click on the Metronome Icon.

Either option will cause the Metronome icon to glow green, meaning it's armed. Right-clicking on the *Metronome* icon displays the Metronome Unify instance, as shown below.

**NOTE:** The Metronome only starts when the transport is playing. If you have the transport set to Manual Start/Stop, then the clock is only started when you click on the Play Triangle and only stopped when you click on the Stop Square. If you use Play modes other than the 2 manual options, then the clock starts when you play a key.

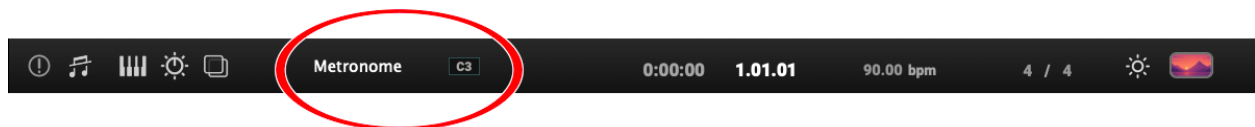


By default, an initialized Unify patch uses a *Metronome* instance that generates a click on ¼ notes. Since there is no On/Off button on the *Metronome* instance UI yet (coming in a future update), we recommend that you turn on the Metronome BEFORE opening the Metronome instance.

When the default Metronome is turned on, playing and holding a note produces a click.

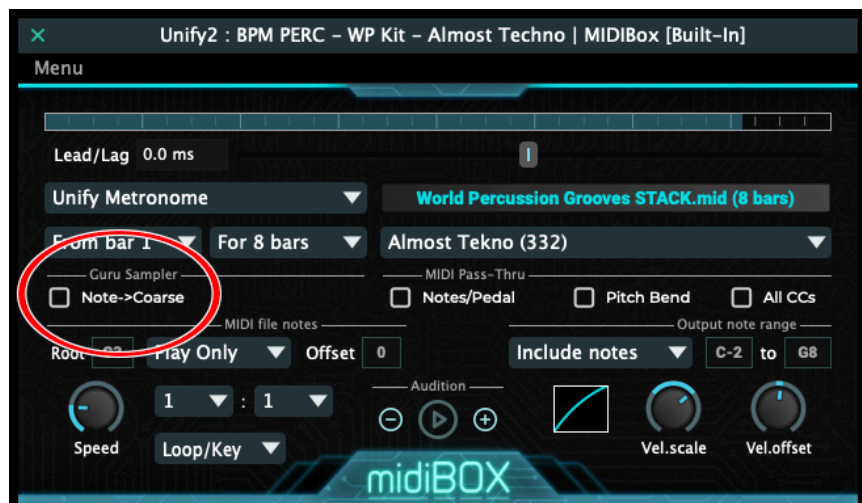
### How to open the Unify 2 Metronome window / instance:

1. Click the chevron above the Metronome Icon and click the “Open Metronome Unify” button.
2. Right+Click or Control+Click the Metronome Icon for a shortcut, single click way to open the Metronome Unify.



The *Metronome Unify instance* has a simplified Transport because it is only a slave to the settings of the main Unify 2 transport. Tempo, Time Signature, even timing advance are not available in the Metronome Unify. There is 1 parameter that is different - to the Right of the “Metronome” text is a note assignment. This is the note that is triggered when the clock starts that creates the ‘Click” or other User-defined metronome sound.

**BONUS TIP:** There is a setting in MIDIBox (our MIDI File Player): “Note -> Coarse” has a special connection to our GuruSampler (V1 and V2) that will change the “Coarse” tuning as you play up and down your keyboard. For Drums this is usually set OFF so the same pitch plays on all keys, BUT with the Metronome Unify, using this



to get different drum sounds by changing the pitch of the samples is a very fun option to explore.

The On/Off State and ALL Settings in the *Metronome Unify instance* are saved in the Unify Patch.

NOTE: The *Unify Metronome instance* runs “side by side” with the main Unify instance, meaning the mix of each is not combined UNLESS you use the “Metronome Mix” parameter, labelled *To Master* in the *Metronome Options Toolbar* to mix the Metronome audio with the Main Unify Master Output. The *Metronome Options Toolbar* is shown when you click the Chevron above the *Metronome icon*.

## Unify Metronome Library

When you open ‘Metronome Unify’ by right-clicking the *Metronome icon*, it opens the metronome unify instance (a parallel instance of Unify 2 dedicated to metronome playback), with the currently loaded metronome patch in it.

The metronome instance browser panel will default to showing the “Unify Metronome” library.

If you are starting with an initialized patch or a patch that was not saved with a custom metronome patch, the default metronome instance will just contain a ‘click box’ in the master layer.

The Unify Metronome library, other than three (3) *Click style* patches at the top, are based on BPM Drum and Percussion Patches. This means the metronome doesn’t have to be just a click, in fact that is probably the LAST thing you’ll want it to be. Most of the factory metronome library BPM patches are driven by MIDIbox. You can create your own BPM patches using MIDIbox, loading your own MIDI files into it, followed by whatever MIDI effects you would like to add, so the timing of the “click” can be as loose as you want it to be.

Taking this concept further, remember the Unify Metronome is a complete Unify instance, where an unlimited number of layers can be loaded into it! Set the loading option to “Insert New Unify” and then click 4 other patches in the Unify Metronome library and you now have a 5-layer rhythmic groove to jam, or record, with. The Metronome ‘groove’ can be recorded by Unify 2’s new built-in recording feature.

We have no idea of all the ways the Unify Metronome can be used. With Audio File players, MIDIBox, Step Sequencer, BlueARP built-in, along with any MIDI effect plug-ins you own, complete songs could be loaded with backing parts that play alongside the main Unify. From live performances to studio productions all can benefit from thinking of unique ways of using the Unify Metronome. The possibilities are endless.

# Layer audio input and output / layer MIDI out

Unify 2 has the ability to pipe audio into Instruments, and the Aux and Master layers, as well as sending Audio to specific hardware outputs.

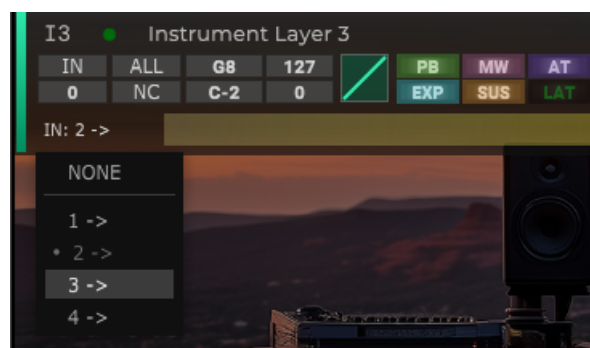
Unify 2 supports up to 16 stereo inputs and outputs, both as a plugin or standalone mode. In standalone mode, this is achieved by selecting multiple inputs and outputs in the audio settings panel.

When using Unify 2 in a DAW, the audio inputs and outputs are accessed via plugin side-chain, input/output options. The way this is done can vary by DAW; please refer to your DAW user manual in that case.

Once enabled, audio inputs and outputs on instrument layers and the aux/master layers appear as selection boxes on the left and right hand side of the layers panel in the full/expanded mode of the panel.

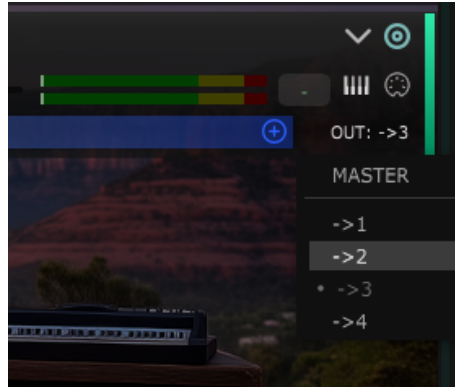
## Selecting an audio input

The available input channels are listed in the order that they appear in the hardware audio input selector (or the virtual inputs available in your DAW).



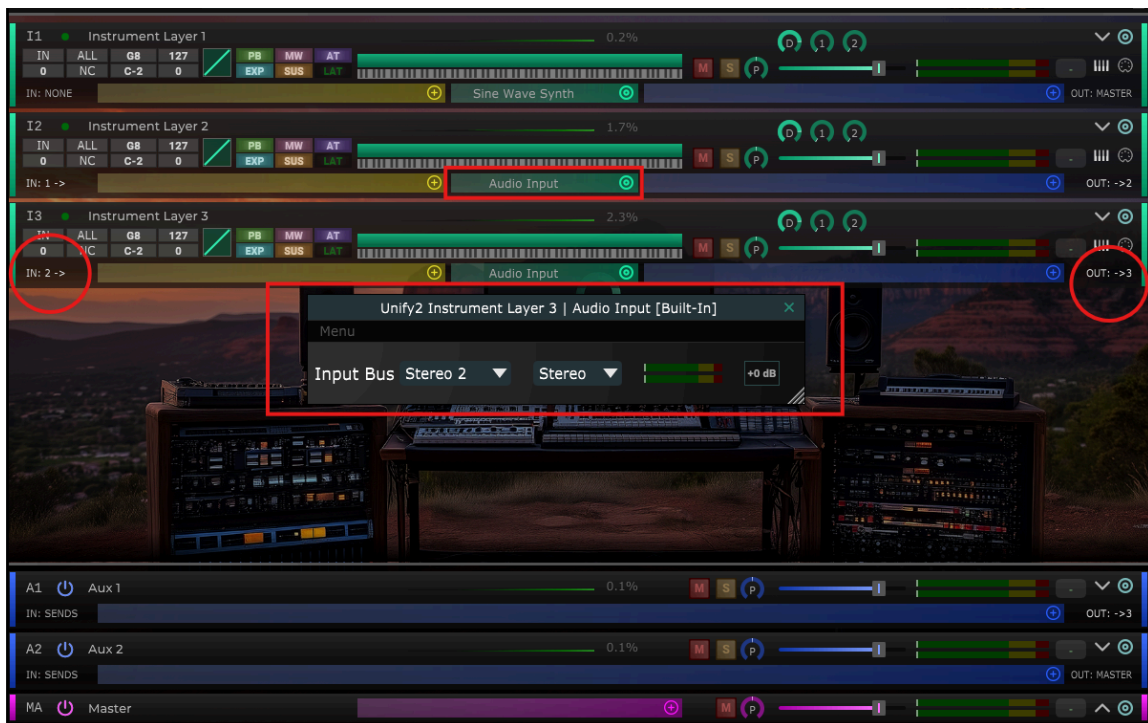
## Selecting an audio output

The available output channels are listed in the order that they appear in the hardware audio outputs selector (or the virtual outputs available in your DAW).



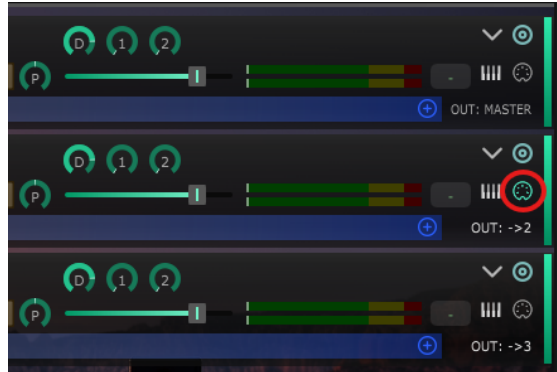
When you select an audio input, if the *Sine Wave Synth* is currently loaded, then it will get replaced by the *Audio Input* plugin, which lets you pick a specific channel and provides a way to view the input level.

If you have a different Instrument loaded, it is not replaced when selecting an audio input. The reason for this is that many plugin instruments also have a side-chain audio input to allow processing sound or using the sound as a side chain to internal effects.



## MIDI out

To enable MIDI out on a MIDI or Instrument layer, simply click the MIDI icon to toggle its state.



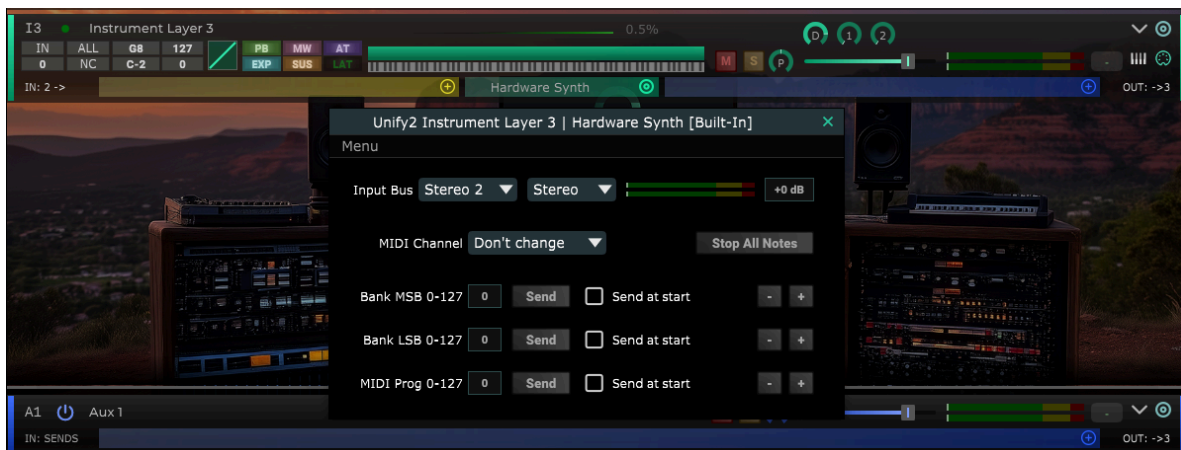
MIDI out passes MIDI data to the selected hardware MIDI output device in standalone mode, or to another track in plugin mode.

MIDI out does not pass MIDI data between layers. If you need to do this, use the MIDI layers and Instrument layers functions that allow a specific MIDI layer as their MIDI input instead of a hardware MIDI channel input.

## Hardware synth

There is also a *Hardware Synth* Unify plugin that enables audio input, whilst forwarding MIDI to the MIDI output port. This lets you set up device specific program changes that will be recalled when the patch is loaded.

We will be adding a delay compensation slider to this plugin soon which may be helpful in some cases if you need to align external synth audio with internal plugins.



# The built in and bundled plugins

There are multiple built in plugins that allow you to use Unify 2 as a fully fledged sound design tool without even needing to use any external synth or effect plugins.

Unify 2 is also provided with some open source bundled plugins for which the authors have kindly given us permission to distribute with the Unify 2 installer.

These 'bundled' plugins are not built into the application binary, rather they are installed in the main application resources folders which allow Unify 2 to always have access to them without needing to do a plugin scan.

Here is the complete list of plug-ins included with Unify, organized by type.

In addition to our own built-in plug-ins, and several bundled third-party plug-ins indicated specifically below, the entire open-source [MDA VSTs](#) package is included, with the kind permission of developer Paul Kellett; see [About the MDA plug-ins](#).

## Instruments

Instrument plug-ins receive MIDI and produce audio.

- Unify 2 itself can be used as an instrument
- Sine Wave Synth (built-in) plays simple sine tones for testing
- Guru Sampler 2 (built-in) our flagship sampler instrument
- [Guru Sampler](#) (built-in) our flagship sampler instrument from Unify 1
- [Audio File Player](#) (built-in) simple audio-file player with MIDI triggering
- [Dexed](#) open-source emulation of Yamaha's classic DX7 FM synthesizer
- [Digits](#) open-source phase-distortion synth similar to Casio's CZ-series
- [mda DX10](#) basic 2-operator FM synth with very low CPU usage
- [mda JX10](#) 2-oscillator analog subtractive style synth, also very easy on CPU
- [OB-Xd](#) open-source subtractive analog-style synth, emulates Oberheim OB-Xa
- [MOrgan Osc](#) open-source tone-wheel organ simulator

## MIDI Effects

MIDI effect plug-ins process MIDI data only (not audio).

- [MIDI Filter](#) (built-in) multi-function MIDI event filter
- [CC Mapper](#) (built-in) flexible MIDI CC filter/re-router
- [CC Rider](#) (built-in) MIDI CC LFO
- [CC Sender](#) (built-in) MIDI CC generator
- [PolyBox](#) (built-in) creates a virtual polysynth from multiple monosynth layers
- [MIDIBox](#) (built-in) MIDI-file player with looping capability
- [JitterBox](#) (built-in) randomizes timing (and other aspects) of MIDI notes
- [TriggerBox](#) (built-in) creative MIDI triggering effect
- [BlueARP](#) sophisticated MIDI arpeggiator
- [Ripchord Player](#) (built-in) turns single key-presses into chords
- [MOrgan Perc](#) Hammond-style, mono-triggered/polyphonic organ “percussion” effect

## Audio Effects

### Delay

- [Omega Delay](#) (built-in) creative stereo delay with integrated distortion and chorus
- [Sample Delay](#) (built-in) sample-accurate stereo delay for latency compensation and special effects.
- [Chow Matrix](#) multi-delay with advanced modulation
- [mda Delay](#) basic stereo delay
- [mda DubDelay](#) creative delay with feedback saturation and time/pitch modulation

### Distortion

- [NoizBox](#) (built-in) creative distortion effect
- [Swanky Amp](#) mathematically modeled tube-amp simulator
- [Chow Tape Model](#) sophisticated reel-to-reel tape simulator
- [mda Bandisto](#) multi-band distortion
- [mda Degrade](#) sample-rate reducer
- [mda Overdrive](#) warm tube-type distortion

## Dynamics control

- [Stereo Gain](#) (built-in) stereo gain/attenuation block
- [Enforcer](#) (built-in) stereo compressor
- [Pump House](#) (built-in) adds tempo-locked pulsation to any sound
- [LoudMAX](#) “brick-wall” limiter
- [mda Dynamics](#) compressor/limiter/gate
- [mda Envelope](#) envelope follower / VCA
- [mda Limiter](#) optoelectronic-style limiter
- [mda MultiBand](#) multi-band compressor
- [mda Splitter](#) frequency/level crossover

## Equalization and Filtering

- [Restrictor](#) (built-in) precision band-limiting filter
- [FilterMax](#) (built-in) single-channel, multi-mode filter
- [FlexEQ](#) (built-in) parametric EQ with high and low shelving filters plus two peaking filters
- [mda Loudness](#) equal loudness contours for bass EQ and mix correction
- [mda RezFilter](#) resonant filter with LFO and envelope follower

## Modulation

- [Silky Chorus](#) (built-in) beautiful clean stereo chorus
- [PhaseMod](#) (built-in) stereo phaser
- [PhaseMORE](#) (built-in) multi-stage phaser
- [FlangeMod](#) (built-in) stereo flanger
- [mda Detune](#) simple up-down pitch-shifting thickener
- [mda Image](#) stereo image adjustment and mid-side matrix
- [mda Leslie](#) rotary speaker simulator
- [mda RingMod](#) simple ring modulator
- [mda RoundPan](#) 3D auto-panner
- [mda Stereo](#) Haas-delay and comb-filtering stereo simulator
- [mda SubSynth](#) multi-mode sub-bass enhancer
- [mda ThruZero](#) thru-zero flanger
- [mda Transient](#) transient shaper
- [mda Tracker](#) pitch-tracking oscillator/EQ
- [mda Vocoder](#) 8/16-band vocoder
- [MOrgan Cab](#) Leslie-effect simulator

## Reverberation and Ambience

- [WaterVerb](#) (built-in) yields a swirling, “underwater” sound
- [HallVerb](#) (built-in) basic hall reverb with Freeze button
- [EmVerb](#) (built-in) high-quality hall reverb with early reflections
- [ZenVerb](#) (built-in) high-quality hall reverb with independent low- and high-frequency timing
- [mda Ambience](#) room-ambience effect
- [Dragonfly ER](#) open-source Early Reflection reverb
- [Dragonfly Plate](#) open-source Plate reverb
- [Dragonfly Room](#) open-source Room reverb
- [Dragonfly Hall](#) open-source Hall reverb
- [KlangFalter](#) open-source convolution reverb by Uli Haberhauer

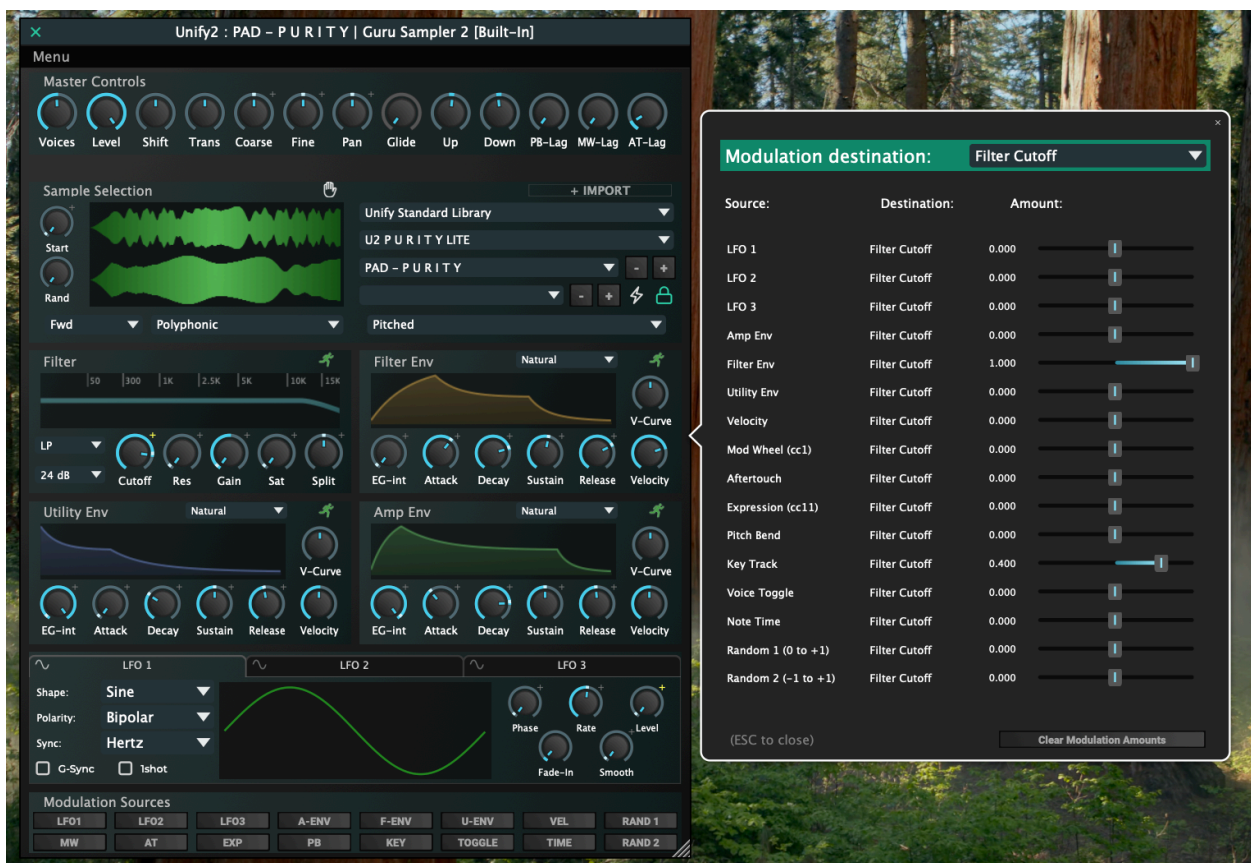
## Special

- [Audio File Player](#) “instrument”
- Audio Analyzer - basic stereo spectrum analyzer and waveform viewer
- [ComboBox](#), which allows connecting multiple plug-ins in arbitrary ways
- [MIDI Monitor](#) MIDI stream monitoring utility
- Audio input
- Hardware synth

# Guru Sampler 2

Whilst it may not be possible to describe every single built in and bundled plugin, Guru Sampler is the cornerstone to Unify libraries.

Guru Sampler 2 in Unify 2 now has become practically a fully fledged modular synthesizer with a very flexible modulation matrix which includes an extra envelope, 2 extra LFOs and polyphonic modulation of all the LFOs by default. Everything is calculated PER NOTE which is something GuruSampler V1 was not capable of. It required a complete ground-up re-design to bring us the incredible power GuruSampler 2 has.

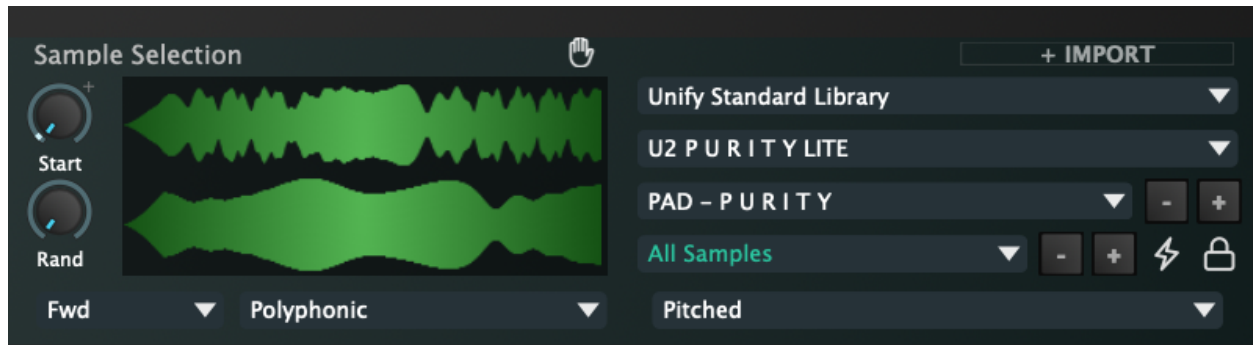


The graphical elements of the UI are now animated to show live modulation and the knobs also now show a live modulation depth amount from the base values.

You can drag the mouse in the filter and envelope panels to adjust the values as well as using the knobs.

In the filter graph, if you right click the mouse whilst dragging you will be able to control the split and saturation amounts.

## Sample panel / Selecting samples



The sample panel lets you pick the multi-sample that will be used - first select the library, then the sound collection within the library, and then the sample (PAD - PURITY in this example).

The 'hand/stop' icon will freeze the sampler to the sample linked to the last key press.

Beneath the sample selector is a pull-down list of samples in the multi-sample - you can also select a specific sample that way.

The graphical waveform view updates on every note event showing the sample being used for that key range.

## Importing Samples

You can import your own single samples using the import button - which will open the sample import panel.

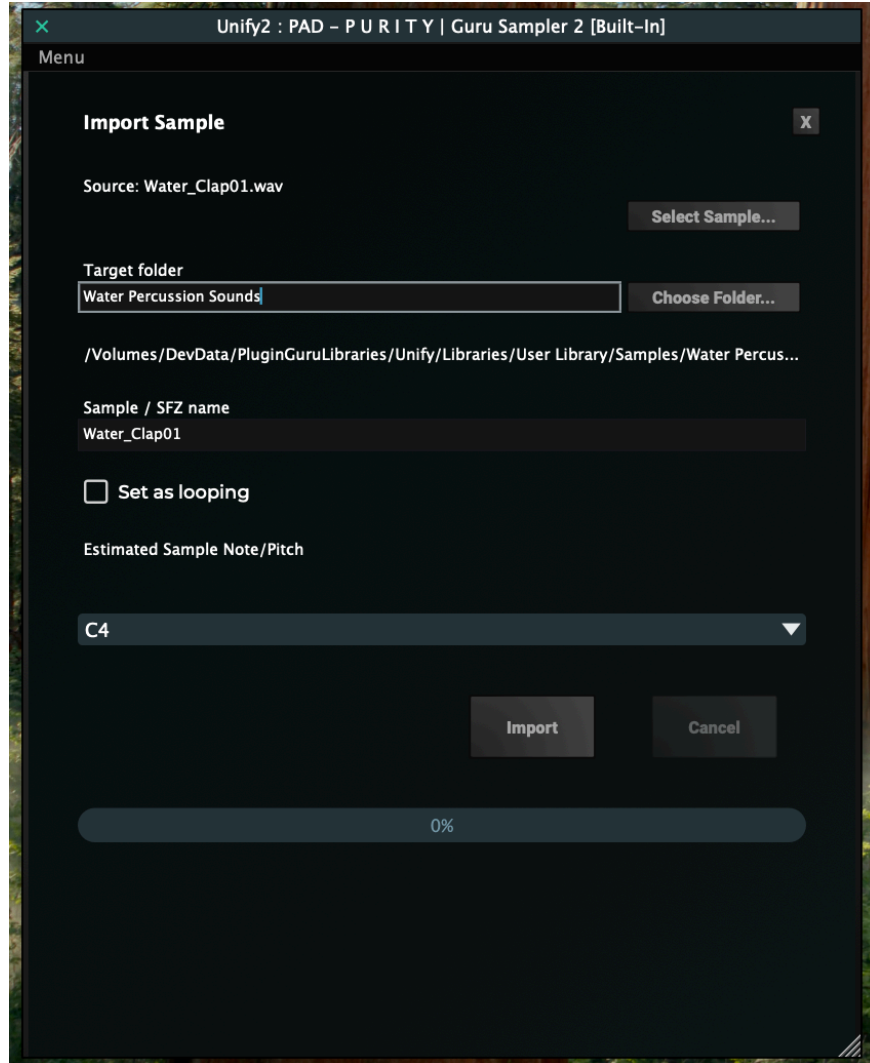
You can either drag and drop files into this panel and click the import button each time, or you can click the 'select sample' button.

Gurusampler import will accept most types of audio file and will convert them to wav format.

The sample importer is a basic importer - if you need to trim/edit your sample audio, you'll have to use an external audio editor, however on importing, the importer will try to trim any silence at the beginning of the sample.

Once a file is set for import the importer will estimate the root note of the sample.

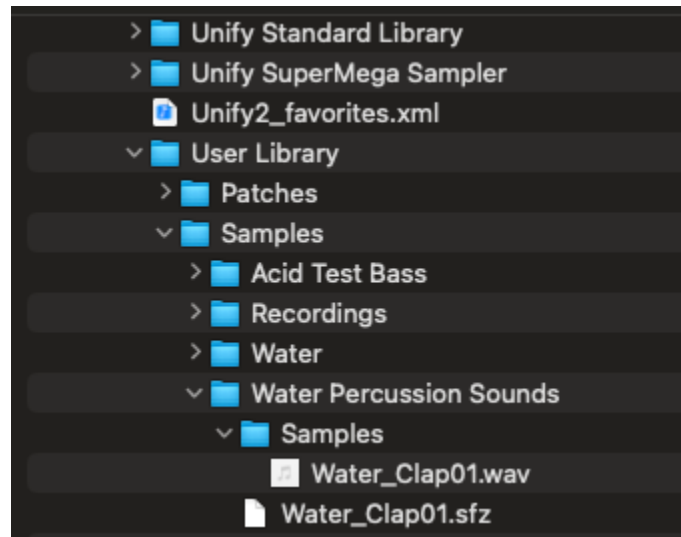
When importing multiple samples of a similar category it's useful to change the target folder name to match how you want it to appear in the sample pull-down list in the main sampler interface.



(sample import panel)

The target folder is where the samples will be imported - giving it a custom name for a collection of sounds is a good idea, and the default location will point to the user library. We suggest importing all your own samples there, but of course you can create new libraries and select those as the import target, just make sure to create a 'Samples' subfolder and use an appropriate 'Target Folder' name for the imported sample/s.

Click the import button and the progress bar will show the progress of the import. The file will be converted to wav 16bit 44.1 KHz and an SFZ file and subfolder with the sample will get created automatically.



(After the import - we have a new sfz and sample in the user library under the newly defined collection name 'Water Percussion Sounds')

To import more samples of the same category you can just stay on the import panel and select new source sample files one at a time - and you can even drag and drop samples into the import panel.

After importing a sample/s you can select it in the main interface - in this case the sample was saved to the user library, in the collection named 'Water Percussion Sounds':



## Filter controls

Gursampler 2 adds some extra filter controls additional to the usual filter type, cutoff and resonance controls.



Saturation - this will drive the signal into a soft clipper before it's sent into the filter, with post filter gain adjustment to keep the level stable.

Split - this will balance the left and right channel cutoff - it's really useful for creating a wider sound from a mono sample (or stereo samples), and generally is some nice 'ear candy' with subtle modulation. Try using it with the 'Toggle' modulation source.

## Envelopes / envelope curves

There are now 3 types of envelope curves to select from as well as the v-curve parameter which lets you adjust the velocity response curve in relation to the velocity amount parameter.



Velocity curve adjustment:



The EG-int parameter can be modulated whilst the voice is active - it will affect the total envelope intensity.

The other envelope parameters are modulation targets but only update their values at the point of the envelope being triggered (when a note-on event triggers the envelope).

## Assigning modulations

There are two ways to assign modulations - either by clicking one of the 'mod source' buttons, or by right clicking a modulation target which causes the modulation mixer panel to open.

When you click on any of the modulation sources, the main panel changes to show which knobs are active targets, and a pie chart modulation depth indicator on each knob shows the amount of modulation applied, in green for positive amounts, and in red for negative amounts.



When you right click on a modulation target knob (there's a subtle + button next to any knobs that are modulation targets), it will open the pop up modulation mixer panel - pre-selected for the current modulation target.



On this panel you can also directly select any of the other modulation targets or access the list of all currently assigned modulations - which is handy if you want to understand how a patch has been set up in terms of the full modulation matrix.

Once a modulation has been applied to a knob, the + indicator changes colour to yellow.

## Patch conversion from Gurusampler 1

It's not necessary to upgrade patches from GuruSampler 1 to GuruSampler 2 unless you really want to access the advanced capabilities of GuruSampler 2, however Unify 2 provides a way to automatically convert patches from GuruSampler 1 to GuruSampler 2.

Gurusampler 1 patches can be converted into Gurusampler 2 patches in two ways:

1 - In the plugin box operations menu in the instrument layer, you can select to swap an instance of the Gurusampler plugin that is already loaded with Gurusampler 2 - Gurusampler 2 will attempt to auto-convert the patch settings and do a best fit of the modulation matrix assignments.

2 - In the patch browser, if you right click on the patch to be loaded, you can select to convert it to Gurusampler 2 - in this case, any layer, even embedded instance layers will auto-convert Gurusampler 1 (GS1) patches to Gurusampler 2 (GS2) and replace the GS1 plugin instance with GS2.

Patch conversion from GS1 to GS2 will select the same samples and do a best effort attempt to copy over all the parameters - but please be aware that because GS2 uses a completely new modulation scheme, some modulation depth settings might not result in an exactly identical sound vs the original GS1 patch - so you may need to tweak modulation settings to get closer to the original sound.

# NoteMate

This is a new built-in plugin introduced in 2.01.0.

NoteMate is a MIDI plugin which allows channel re-assignment of incoming notes in 3 keyboard zones.

Each zone has a voice matrix of 8 virtual voices that it uses to allocate incoming notes.

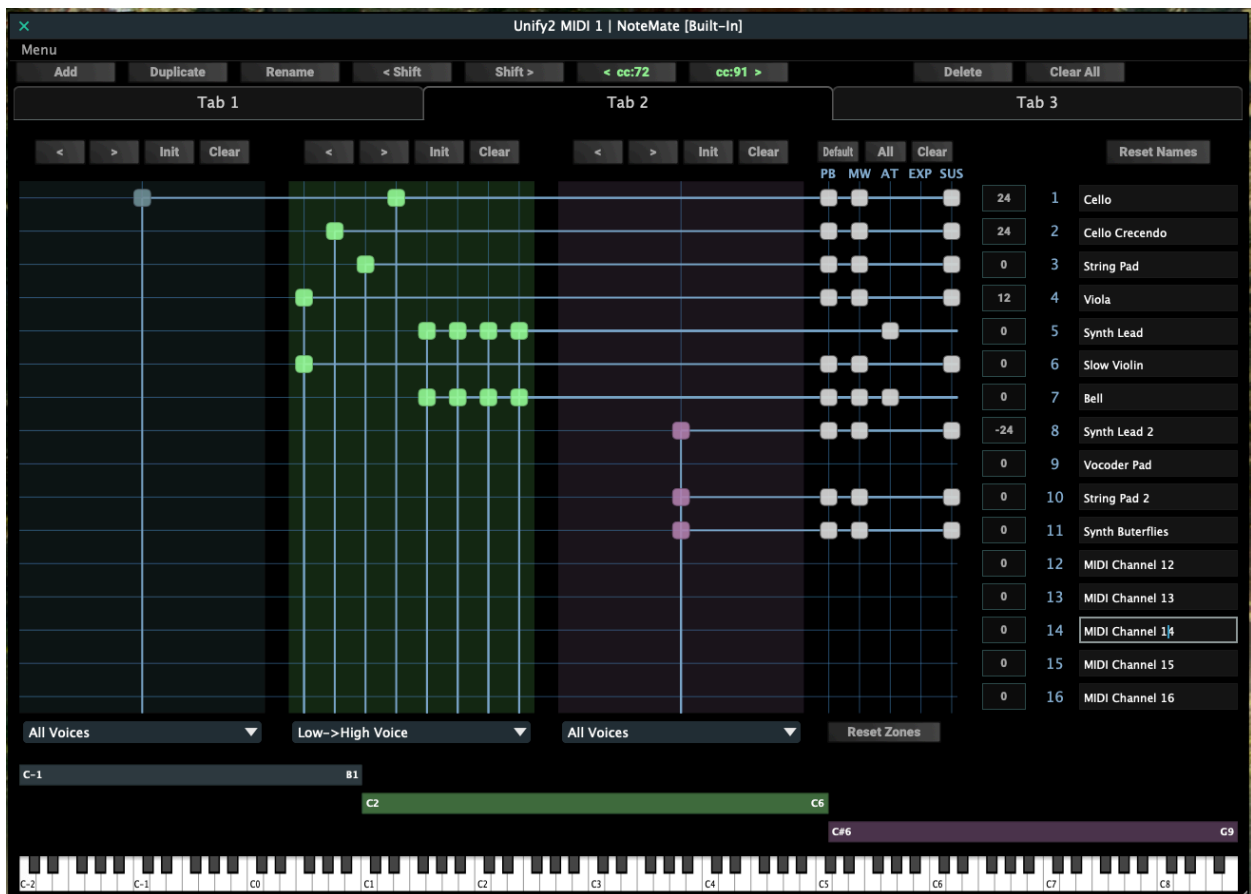
Voice ordering is from left column to right column in each zone - voice 1 to 8.

In the default low to high allocation mode, each new note held down is assigned to play to a voice - from left to right.

In this example screenshot - the very first note in the middle zone will actually generate a note to channel 4 and 6, the next note pressed/held will go to channel 2, then channel 3, then channel 4. The 5th and consequent held notes go to channels 5 and 7 in this case.

In this example there are 3 tabs. Each tab can have a different matrix configuration and you can rename the selected tab with the rename button.

Channel names can be given custom names, and these names will be the same in every tab - since the target instruments in Unify2 will be the same for each channel.



To the right of the note zones there are matrix check boxes for each channel to allow passing through or blocking of specific MIDI controller messages.

PB = Pitch Bend,

MW = Mod Wheel (cc1),

AT = Aftertouch (channel and poly aftertouch messages),

EXP = Expression (cc11)

SUS = Sustain pedal.

The number boxes to the left of the channel numbers allow you to set a different transpose amount for each channel per tab.

NoteMate allows you to switch tabs during live performance by selecting the tab, clicking the next/previous button - or you can assign a CC controller to those buttons by right clicking and then moving the controller - it will be picked up and displayed on the button.

You can also remote control the tab selection via parameter links that are exposed to the Unify2 Macro controls parameter links editor/modulation panel.

Other modes for voice allocation:

High to low:

Voices will be allocated from right to left column - so the first note received/held will go to column 8/voice 8, and next notes will be 7,6,5 etc.

Random:

Voices will be allocated randomly - so the selected column on each note down will be random.

All Voices:

In this mode there is only one column - all notes received go to the checked channels - so you can send the same notes to multiple channels - but you can still do per channel transposes and MIDI control messages filtering per channel in this case.

Zone range editing:

You can drag the zones directly with the mouse or you can right click to enable note learning to set the top/bottom note of a zone.

## Using NoteMate in a Unify2 patch:



To use NoteMate as intended in Unify2, you should put it in a MIDI layer - usually MIDI layer 1. Then you can create multiple instrument layers - each layer should be set to listen to the MIDI layer in which NoteMate is placed.

A handy way to create those layers is to set up 1 instrument layer with the correct settings for listening to MIDI layer 1 and MIDI channel 1, and then click the actions/target button on the layer to 'Dup 7x next MIDI ch' (for 8 channels total for example) - or however many channels you want to add.

Important note about Polyphonic Aftertouch - you need to be aware that if you transpose notes in NoteMate or on the layer before notes reach a plugin supporting poly aftertouch, the aftertouch messages will not work as expected because the note numbers will no longer match the MIDI note linked to each poly aftertouch message - so do the transpose of the pitch directly inside the plugin.

# Step Sequencer

Step Sequencer is a built-in MIDI plugin introduced in Unify 2.00.0.

It's a very simple sequencer - designed to be easy to use for setting up basic pitch offset sequences with variable velocity and gate times.



The step sequencer works in two modes - using MIDI input - it replaces each note received with an output note at the relative pitch offset of each step in the sequence - so you can play a chord, and the chord will transpose on each step, or you can play a single note.

You can disable MIDI input to let the sequencer run at a fixed root note - selectable by the note selector box that appears next to the 'USE MIDI IN' option when it's un-checked.

In this mode, the step sequencer will output only one note at a time and will not change pitch offset in relation to any incoming notes.

The steps movement is determined by the transport - so the transport must be running for the step sequencer to play through the steps.

## Step editing:

There are 3 rows of main step controls - the pitch - with the blue bars also representing the gate time in the step, the gate % for each step (gold/orange bars), and the velocity level (green bars). When 'scale velocity by received notes' is enabled, the output velocity of each step is multiplied by the relative MIDI input velocity of the received notes.

The screenshot displays a step sequencer interface with 16 steps. The top section includes controls for Transpose (0), Steps (16), Continue Steps (0), and a range selector (1 to 16). A central menu lists 36 scales, with '07 - Aeolian' selected. Below the menu, the sequencer shows three rows of data for steps 1-7 and 12-16: pitch offsets (e.g., +7, +3, +5, +7, +7), gate percentages (e.g., 99%, 48%, 75%, 28%, 41%, 54%, 31%), and velocity levels (e.g., 1.00, 0.72, 0.61, 0.80, 0.76, 1.00, 0.98). The bottom section features a 'Learn Steps' checkbox, 'Scale Lock', 'Root: C', and 'Scale: 07 - Aeolian'.

Selecting a scale will force the output notes to be quantized to the nearest note relative to the root note and scale notes. You can select the custom scale option, and then edit the output scale by clicking on the note dots to the right of the scale list.

There's a handy note/pitch input mode: When the 'Learn steps' checkbox is enabled, you can press a note on your keyboard to set the starting note, and consequent notes entered will define the pitch offset in semi-tones for each step - auto advancing the step all the way through the available step count that has been defined.

When you are done inputting pitches/steps you can un-check the 'Learn steps' checkbox to get back to normal step sequencer playing mode.

## Per step buttons:

Underneath each step there are 3 buttons - the circle is a step enable/disable button, the T button is to enable a note tie from the current step to the next, the R button is the 'Random next step' option - this will make the sequencer jump to a random step when this step is reached.

## Keeping the sequence running:

The step sequencer will keep playing notes whilst a key is held down in MIDI IN mode, but will stop when keys are released. You can keep the sequencer playing by holding down the sustain pedal. Typically you don't want the sustain messages to pass through to the synth that is playing a fast sequence, so by default 'Enable sustain pedal pass through' is disabled.

Another way to keep notes going after keys are released is to set a number of 'Continue Steps' - when keys are released the steps will continue for the set number of steps, providing the transport is running (so this won't work in quick stop mode - but is useful in manual or MIDI 1,2 etc bar modes).

When MIDI input is turned off, the step sequence will play automatically so long as the transport is running, starting at step 1 when the transport is started.

## Timing options:

Other options at the top area of the step sequencer allow you to define the total number of steps and the currently playing step range, as well as the timing of each step - in beat time relative to 1 bar of quarter notes, or as a time based on whole bars - for really long note transitions suitable for things like Ambient drones or for maybe using a step sequencer as the note input to another step sequencer to have a slowly changing root input note.

The transpose slider affects the output of all notes by pitching up or down by the set number of semitones.

# Appendix / Advanced considerations

## Keyboard shortcuts

Here is a list of keyboard shortcuts available in Unify. Keyboard shortcuts are quick ways to access alternative functions by holding down one or more “modifier keys” on your keyboard while clicking a GUI element with the mouse.

### General

#### The mouse:

- Most computer mice have at least two buttons: left and right.
- “Click” (aka “single-click”) means “left-click” unless otherwise specified
- “Right-click” means to click with the right button instead of the left one
- “Double-click” means click the left button twice in rapid succession.
- “Drag” means press the left button down *and hold it down* while moving the mouse.

#### Modifier keys aka “shortcuts”:

- ALT is called “Option” on the Mac; CTRL and SHIFT are the same on both Mac and PC
- We don’t use the Mac CMD (“command”) key, because the corresponding “Windows key” on a PC is technically equivalent, in practice it is hijacked by Windows to open the Start Menu, and hence not available for other uses.
- We don’t distinguish between the SHIFT/ALT/CTRL keys at the left and right ends of the keyboard.

**Ordinary single-clicking** (without holding down any key) is used to select items in a displayed list, activate an on-screen button, or to open a pop-up menu.

- It is even more commonly used as the start of a “drag”, where the left button is held down while the mouse is moved, e.g. to adjust on-screen knobs/sliders, or re-order plug-ins on a layer.

**Right-clicking** (clicking the right mouse button instead of the left one) is always a shortcut to the “most common” operation, e.g.

- Right-clicking on menu icons (i.e., right-clicking in a place where a left-click would open a menu) selects the most common menu choice, without even displaying the menu.
- Right-click operations are always harmless, and usually open some kind of pop-up window.

**CTRL + Click** (hold down the CTRL key while left-clicking) serves as an alternative to “right click” because

- Many Mac users have a traditional single-button mouse, and
- Users with limited hand mobility find the right-click operation difficult

**ALT-click (PC) or Option-click (Mac)** is most commonly a “power-user shortcut”

- On menu icons, selects a non-harmless operation, e.g. removing a loaded plug-in.
- If you're not absolutely sure what holding down ALT/Option will do, check this list first!

### Third-party plug-ins

- The shortcuts described here apply only in windows which are created by Unify itself, e.g.
  - The main Unify window
  - Linked-parameter windows
  - *Built-in* (as opposed to *bundled*) plug-ins
- *Third-party plug-ins* (including all the *bundled* ones like *OB-Xd*, *Digits*, etc. have been programmed by other people, and most have their own entirely different shortcut conventions, over which we have no control.

### INIT button

- **ALT + Click** the INIT button to delete all INST layers except the first
  - (Useful when working with [PolyBox](#))

### Selecting patches in the patch browser

The upper half of the patch browser is called the *selector*.

- **Right-click** or **CTRL + click** the “...” button beside the library-menu, to access the list of available [library subset presets](#) without having to first go to the [Library Selector](#).

The lower half of the [patch browser](#) is called the *chooser*.

- **Click** (no modifiers) any patch to load that patch, replacing any previously-loaded patch
- **Right-click** or **CTRL + click** any patch to pop up a menu of available operations (shortcuts below provide quick access to some of these)
- **ALT + Click** any patch to add its layers to the current patch
  - Not all layers will be added
  - The new patch's Master Effects layer will never be added
  - INST layers will be added
  - **CTRL + ALT + Click** adds ONLY the INST1 layer to the current patch.
- **SHIFT + Click** any patch to load it as a new “Unify layer” (INST layer with Unify as the instrument)
  - **CTRL + SHIFT + Click** is the same, but automatically suppresses pitch-shift settings in any [MIDIBox: MIDI file player](#) instances (useful for drum grooves)

After clicking any patch to load it:

- The **cursor-down key** on your computer's keyboard selects and loads the *next* patch in the chooser
- The **cursor-up key** selects and loads the *previous* patch
- The **F** key toggles the “heart” icon (favorite status indicator) for the *currently-loaded patch*

## Ops (operations) buttons

“Ops” (operations) buttons, displayed as a small icon of two concentric circles, are used throughout the Unify GUI; clicking them always pops up a menu of available operations for the part of the GUI in which the ops button appears. Holding down ALT (Option) when clicking most ops buttons provides a shortcut to selecting a specific menu item. (I.e., these are true shortcuts. Every function accessible by using ALT/Option is also available on the ops menu.)

## Plug-in ops buttons

In Unify's [layer-stack view](#), plug-in instances are indicated by colored boxes containing the plug-in name, an ops-button on the right and (effects only) a “bypass” button on the left.

- **ALT + Click** to immediately delete the plug-in (effects only)
- **SHIFT + Click** to paste plug-in state from clipboard
- **CTRL + SHIFT + Click** to copy plug-in state to clipboard
- **SHIFT + Click** to paste plug-in state from clipboard

## Layer ops buttons

A *layer ops button* appears at the far right end of each layer, and its operations apply to the whole layer.

- **ALT + Click** to immediately delete the layer

## Curve editors

In any “piecewise curve editor”, where a custom curve can be built from multiple segments (e.g. velocity curve editors):

- **ALT + Click** and drag to drag a control point up/down only (left/right position locked)
- When vertical and/or horizontal grid lines are present, holding down **SHIFT** while dragging control points activates *snap-to-grid*.

## Number boxes

A *number box* is any small box in which a number (e.g. -12), or note-name (e.g. C#5) is displayed.

- **Right-click** (or CTRL + Click) to pop up a “value editor”
- **ALT + drag** in any *transpose box* to adjust up/down by 12s (octaves)

## Parameter sliders/knobs

*Parameter sliders* include most kinds of sliders and knobs available in Unify, including layer level, pan, and aux-send controls, and most parameter knobs in *built-in* plug-ins.

### Layer level sliders

- **ALT-drag** any one layer level slider to adjust all others by the same relative amount
  - “all others” = level sliders for all INST and AUX layers (if any)
  - “same relative amount” = same number of dB

### Macro parameter knobs

- **Right-click** (or CTRL + Click) any knob or its ops button to open the linked-parameters window for that knob
- **SHIFT + CTRL + Click** “the ops button to Copy the knob's name and links to the clipboard \* SHIFT + Click” the ops button to Paste from the clipboard to another knob
- **ALT + Click** selects the “INITIALIZE” menu item, clearing all linked parameters for that knob, and removing any user-assigned name.

### Parameter knobs in built-in plug-ins

In all of Unify's Built-in plug-ins, you can hold down CTRL while adjusting a knob to make fine adjustments.

### Virtual Keyboard view

The Virtual Keyboard (Show MIDI) view displays colored sliders to indicate the state of the MIDI pitch-bend and up to four selected MIDI CCs (defaults 1, 2, 4, 67).

- **Click/drag** in any of these to send the corresponding MIDI messages into Unify
- **Right-click** (or CTRL + Click) in any CC indicator to pop up a menu of available MIDI CCs (with standard descriptions where applicable) to change the corresponding CC number.

## Multi-selection lists

A *multi-selection list* is any scrolling list in which it is possible to select (highlight) one or more items. The main examples are in the [Plug-in Subsets view](#) and the top half of the [patch browser](#).

- **Click** (no modifiers) to select an item, and simultaneously de-select all others
- **CTRL + Click (CMD + Click on Mac)** to toggle an item's selection
- **SHIFT + Click** to select a *range* of items (click 1st item, Shift+click last in the range)

## OSC support

(Note that as OSC support has not changed in Unify 2, vs Unify 1 this section is extracted from the Unify 1 user manual)

[Open Sound Control](#) (OSC) is a network-based protocol for remote control of many types of software and hardware (primarily sound and lighting equipment). Unify has had built-in support for OSC for a long time, but this was only exposed in release versions in Unify v1.7.

Unify's OSC support remains *highly experimental, subject to change, and essentially unsupported*. OSC itself is a bit complicated, and really only suitable for technically-inclined users. If you are interested in experimenting with OSC in Unify, refer to the [PlugInGuru User Forums](#). This manual page provides only minimal details.

### Overview

Unify v1.7 and later can be configured to listen for OSC connections on a specified UDP port; see “Configuration” below.

**Only the Unify stand-alone Unify app can use OSC;** the plug-in versions cannot.

Unify uses only UDP for OSC connections; it cannot use TCP at all.

At present, it basically only receives OSC messages; its support for sending them is extremely limited.

### Enabling and Configuring OSC

To set up OSC support in Unify v1.7 and later, go to the [Settings view](#) and locate the OSC section:



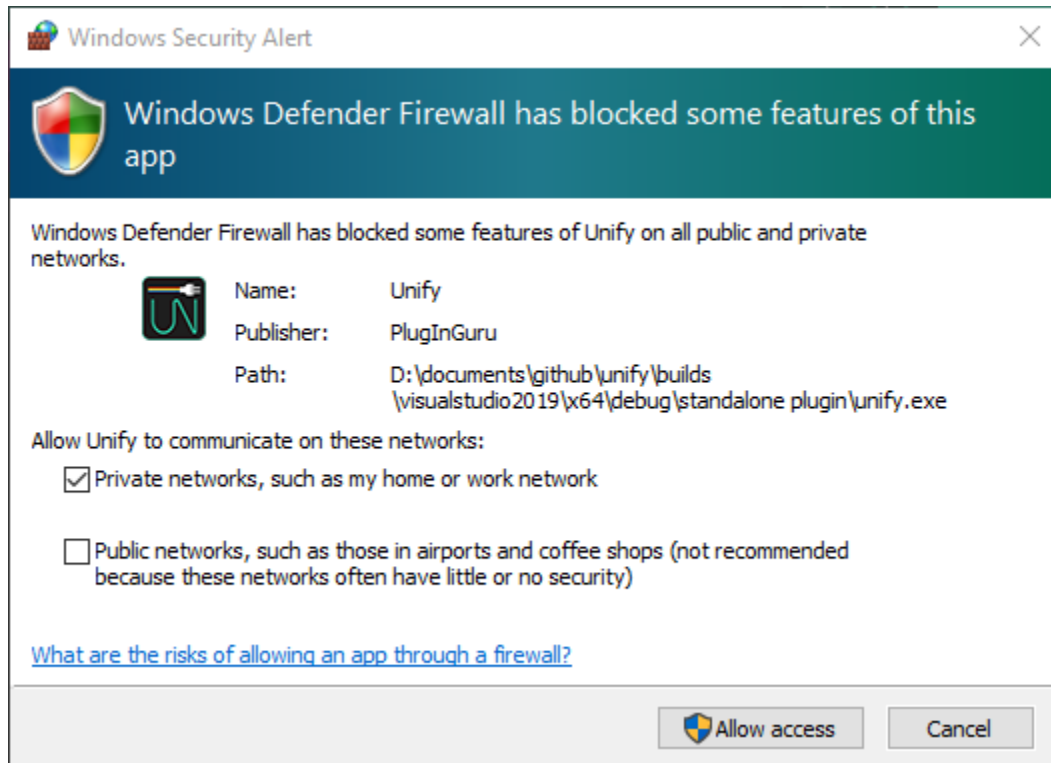
To enable Unify to listen for OSC commands, the least you need to do is to check the “Enable” box.

If you wish, you may also adjust the *portNumber* and/or *ipAddress* settings. **You must *disable OSC* (un-check the “Enable” box) before editing these values, then *re-enable OSC* (check the box again) to accept the changes, and finally *re-start Unify* for the changes to take effect.**

- The default *ipAddress* “0.0.0.0” (shown as “ANY IP”) configures Unify to listen on ALL available network interfaces.
  - If you want it to listen only for *local* connections, from other programs on the *same computer*, change this to “127.0.0.1”.
  - If you want it to listen on a specific interface only, enter the IP address to use.
- On most computers, the default *portNumber* 9001 should not interfere with any existing network services.

- If you need to change it, you'll need to determine a suitable port number, between 1024 and 65535.
- Changes you make will only take effect after you quit Unify, then restart Unify.

**On Windows**, the first time you restart the Unify stand-alone app after enabling OSC, a pop-up window will appear, asking you if it's OK to allow network access:



The recommended setting (allow Unify to communicate on Private networks only) should already be selected. Click the **Allow access** button to confirm it.

## OSC message types

OSC messages take the form of a “path” and some associated data. At present, Unify can handle three kinds of OSC messages:

1. **Parameter-set messages** use a path exactly as used in Unify's linked parameters (see [Using Unify's Macro Knobs for real-time control](#) and [Parameter paths reference](#)), but with a single slash character / prepended at the start.
  - The associated value (to which the parameter is set) *must* be a floating-point value (format-code “f”). No other data types are supported at this time.
2. **Command messages** have the same structure as parameter-set messages, but instead of setting actual parameter values, these trigger *actions* based on the path.
  - All command messages include a floating-point value, but the value itself is ignored for most commands.
3. **MIDI-over-OSC** messages use the path `/midi` and the non-standard format code “m”.

## Parameter-set messages

All parameters which can be linked to Unify's [Macro Knobs](#) can also be set via OSC messages. The paths are almost the same, but the OSC versions have a single slash-character (“/”) added at the beginning.

Note that **Unify's parameters ONLY accept floating-point values between 0.0 and 1.0**. Many parameters (especially plug-in parameters exposed for host-automation) have an *apparent range* which is different, e.g. 0 to 100%, -60 to 0 dB, etc., but internally, the plug-in always maps between the “normalized value range” 0.0 to 1.0 and these “human-readable” values. It is up to you to determine what normalized values to use in OSC messages.

## Command messages

Unify also supports OSC “commands” which are basically like pseudo-parameters, but instead of identifying a specific parameter to be set, the path specifies a specific *action* to be triggered.

Unify v1.7 supports the following “command paths”:

- `/init` is the same as clicking the INIT button.
  - The associated floating-point value must be present, but is ignored.
- `/load` triggers loading of a patch.
  - The associated floating-point value is cast to an integer, and should be equal to the value of the **id** field for the desired patch's row in the Unify patch database.
  - If the value does not exactly match a valid patch ID, the command will be ignored.

- `/embed` also triggers patch loading, but the specified patch is loaded as a new Unify layer.

## Bidirectional messaging for patch-loading commands

As described above, the `/init`, `/load`, and `/embed` commands allow a remote-control program to trigger loading of patches. Because this is often a time-consuming operation, it's desirable to have a way for Unify to let the remote-control program know when the load operation completes. Unify supports a very limited bi-directional messaging protocol for this purpose.

A remote-control program can *optionally* send the OSC message `/hello/IP address` to Unify, with an integer value (format-code "i"). The IP-address string indicates the address on which the remote-control program itself is listening for OSC messages, and the integer value indicates the port-number. An example might be `/hello/127.0.0.1` with value (port number) 9000.

If Unify has received a valid `/hello` message, it will acknowledge the completion of any patch-load command (including `/init`) by sending an OSC message `/presetLoaded` to the specified UDP address and port number. This message includes an integer argument (format-code "i"), whose value is always zero. *Note that Unify will also send these messages even when patches are loaded manually by the user, through the GUI.*

Before shutting down, a remote-control program which has previously sent a `/hello` message should also send a `/goodbye` message, to indicate that Unify should no longer send out `/presetLoaded` messages. The `/goodbye` message also requires an integer value (format-code "i"), but the actual value is ignored.

## MIDI-over-OSC messages

MIDI-over-OSC message formats are not standardized in OSC. The format Unify uses is based on that used by [TouchOSC](#). C++ code to create such messages (using the [TinyOSC library](#) looks like this:

```
void OscSender::sendRawMidi(uint8 byte1, uint8 byte2, uint8 byte3)
{
    if (oscPort < 0) return;

    uint8 data[4];
    data[0] = 0;
    data[1] = byte3;
    data[2] = byte2;
    data[3] = byte1;

    int len = tosc_writeMessage((char*)oscData, sizeof(oscData), "/midi",
    "m", data);
    while (!socket.waitForReady(false, -1)) Thread::sleep(1);
    socket.write(oscAddr, oscPort, oscData, len);
}
```

Note the reversed byte-order, and how 3-byte MIDI messages are encoded as 4-byte integers. (2-byte MIDI messages are encoded the same way, but the `byte3` value will be 0.)

## OSC messages to embedded Unify instances

**As of Unify v1.9**, it is possible to direct OSC commands to specific embedded Unify instances. The required path syntax is similar to that for sending parameter-update messages to plug-ins, but the word *unify* is used instead of *plugin*, and this is followed by the OSC command to be sent to the Unify instance.

For example, to send the command `/master/mixLevel` to an embedded Unify instance on layer INST1, the full OSC path would be `/inst/1/unify/master/mixLevel`.

# Key Zones

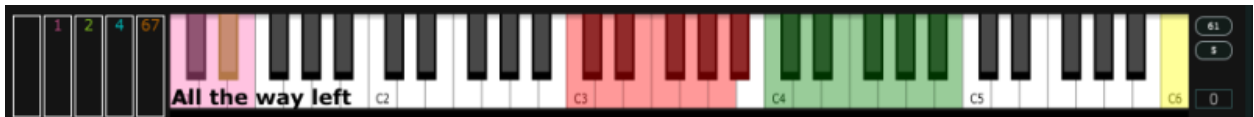
Unify supports the concept of *Key Zones*, where specific note-ranges can be highlighted with colors on the Keyboard panel in the lower area of the Unify GUI, like this:



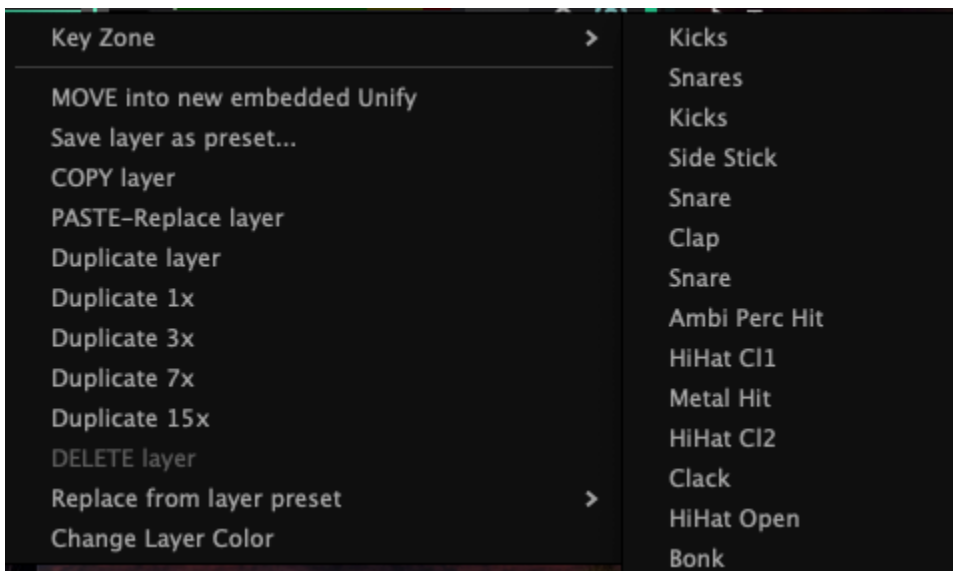
Individual zones have names, which appear when the mouse is hovered over the zone, as shown below:



Unify will display the zone-name text centered within the zone if possible, or shifted to ensure that the full name is visible, like this:



When a patch has key-zones defined, the **ops menu** for any INST or MIDI layer will include a “Key Zone” sub-menu, from which you can choose the name of any zone, to quickly set that layer's key-range to correspond to that zone.



---

**Note: the remainder of this page describes an advanced topic. You must be comfortable with creating and editing plain-text files to be able to use this technique.**

---

## Defining key-zones using a text file

To define a group of key-zones, to which any patch in a library may optionally refer (see below), you must create a plain-text file (extension `.txt`) in the *Key Zones* folder for that library. (If no *Key Zones* folder exists, you will need to create it.)

```
# This is a comment. Comments and blank lines are ignored.
# extra space before, after, and between elements also ignored.
# Each non-blank, non-comment line defines one zone. Format is lokey hikey
color name, where
# lokey/hikey can be note numbers or names
# color can be
# 6 hex digits RRGGBB
# or a standard color name like "pink"
# 8 digits AARRGGBB (to specify per-color alpha)
# optionally followed by blank space and a decimal alpha value
e.g. 0.5
# name is everything from 1st nonblank character after color, to end
of line

# If you want to use note-names instead of numbers, include a line like
this:
    Middle C is C3
# (just has to contain the word "middle" and any digit)

# Default alpha is 0.5, but if you want something different, include e.g.
Alpha 0.6

D0 e1      hotpink 0.35   All the way left
c3 a#3     #1100f0       Zone A
C4 B4      aaFF0080      Middle Zone
96 106     yellow        All the way right

# Optional: custom key-down colour instead of default green
keyDown lightcoral

# Optional: custom mouse-over colour instead of default yellow
mouseOver pink 0.5
```

Basically, the file is a sequence of lines, each of which defines one zone. Each zone is defined by its start and end note, its color, and its name.

- The *start note* and *end note* can be specified either as MIDI note-numbers (range 0-127) or using note names like C#4 or Bb6. (See below for details.)
- The *color* can be specified either as a 6-digit hexadecimal color (format *rrggbb*, with our without a leading “#”) or using a standard HTMLcolor-name. (See e.g. [https://www.w3schools.com/colors/colors\\_names.asp](https://www.w3schools.com/colors/colors_names.asp).)
  - 6-digit hex strings and standard color-names will have an implied alpha (transparency) of 0.5. You can change this default by adding a line like “Alpha 0.6” as shown.
  - To specify the *alpha* (transparency) for an individual color, either use a standard 8-digit hex string (format *aarrggbb*), or add a decimal fraction like “0.6” after the color (separated by a space).
- The *name* runs from the first non-blank character after the color, to the end of the line.

MIDI note-numbers are unambiguous, but are not as human-friendly as note-names like C#4. Unfortunately, such note-names are inherently ambiguous, because there are multiple conventions for the octave-number part:

- Apple/Logic Pro uses C3 to mean MIDI note #60 (middle C)
- Roland uses C4 instead
- FL Studio uses C5

Because of this, Unify allows you to put a line like “Middle C octave number is 3” into your zones text-file, to specify which convention you’re using.

- Any line which starts with the word “middle” (possibly preceded by blank space, case not significant) and at least one digit will be interpreted this way. Unify will use the first digit it finds on the line; **it must be 3, 4 or 5**.
- If you don’t include such a line, Unify will use the middle-C octave selected in the Settings view as the default, but *you should always specify the octave-numbering convention if you intend to share your zones file with anyone else*, because not everyone uses the same setting.

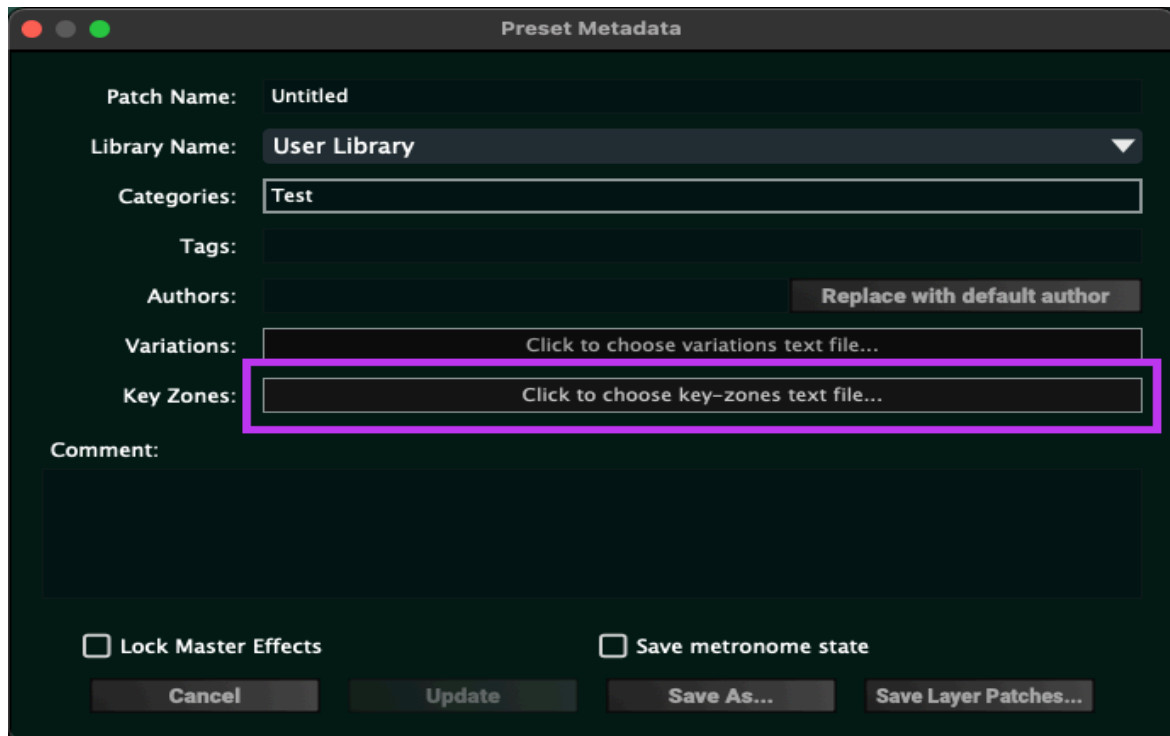
If your chosen zone colors are such that the default key-down highlight color (green) and/or the default mouse-over color (yellow) don’t provide enough contrast, you can optionally specify these colors as shown above, with or without alpha. (Opaque colors are recommended for key-down, semi-transparent colors are best for mouse-over.)

Finally, you can include any number of *blank lines* and *comment lines* (starting with “#”), and use spaces and/or tabs however you wish, to make your file more legible. If you specify a custom “key-down” color, it will also be used for the sustain-pedal-down indicator bar in the [Show MIDI view](#).

## Selecting a key-zone file when saving a patch

The patch-save dialog in Unify 2 includes two new buttons as shown below.

- If you're creating the very first patch in a new library, **save the patch without selecting a zones or variations file first**, to create the necessary library folder. **You will then have to create the *Key Zones* sub-folder manually**, in the *Mac Finder* or *Windows Explorer*, and put your zones file into it.
- Click the “Key Zones” button to select the key-zones file you created from a pop-up menu:
  - The first item allows you to *Clear* the key-zone file selection (if you decide you don't want to use one)
  - After this, you will see a list of all *.txt* files in your library's *Key Zones* folder, from which you can select.
  - Below, you will see a list of *.txt* files in the *Key Zones* folder of the *Unify Standard Library*, which you may also select.
  - If you hold down *Option/ALT* when clicking the button, there will also be an *Open...* item near the top. This will pop up a standard file-open dialog, so you can navigate to any location to find the zones file you want. *Note this will make your patch non-portable to other machines.*



Note: If you have already created the *Key Zones* folder and zones file(s) within it, you can load a zones file without saving your patch if you wish. Click *Save*, choose the zones file as described above, and then click *Cancel* to close the patch-save dialog without actually saving.

# Parameter paths reference

**Note: this information is subject to change in future versions.**

Unify uses a systematic naming scheme for identifying parameters, which is based on “parameter paths” used in OSC. As in OSC, paths consist of a sequence of *components* separated by slash (“/”) characters, but unlike in OSC, there is no initial “/” at the start. (See also the Appendix section ‘OSC Support’.)

## Basic concepts

### Paths, Components, and Values

Each path distinctly identifies a parameter, which may be either a parameter of some plug-in instance, or something internal to Unify, that is to be changed. Changing a parameter can be thought of as sending a “message” consisting of the path (which identifies *what* is to be changed) plus a “value” (which specifies *what it is to be changed to*).

Presently in Unify, the value will always be a *fraction*, i.e., a real number between 0.0 and 1.0 (or if you will, 0 to 100%). The *parameter receiving the value* may interpret it in various ways, however, e.g.:

- A filter cutoff parameter might map the range 0.0 - 1.0 to something like 0-20000 Hz
- An “on/off” parameter like mute or solo might treat anything below 0.5 as “off” and anything above as “on”
- Many plug-ins have “choice” parameters having a finite number of permissible values. These are often selected in the plug-in's GUI using a menu or a knob with fixed positions. Fractional input values are typically interpreted by dividing the total range 0.0 - 1.0 into equal segments, e.g. a synth might offer a parameter for controlling how a pair of filters are to be connected, with choices “off”, “series”, and “parallel”. Input values below 0.33 might be interpreted as “off”, between 0.33 and 0.67 as “series”, and above 0.67 as “parallel”.

### Path validity

Most parameter paths in Unify refer to specific layers using a layer-number component, and many refer to individual plug-ins on the selected layer using an index-number component. Because the number of layers and plug-ins varies from one patch to the next, a given path might or might not correspond to an actual plug-in, and even if it does, the specific plug-in parameter named in the path might not be a valid one for that plug-in. Hence a path may be *valid* in some contexts, and *invalid* in others. Invalid paths are harmless; they simply don't do anything.

## Path Categories

Unify's parameter paths fall into three categories, based on the first component:

1. Top-level paths begin with “unify”
2. Macro paths begin with “macro”
3. Layer paths begin with “midi”, “inst”, “aux”, or “master”

## Top-level paths

Starting with Unify v1.6.0, the syntax of “top-level” paths has been reworked to be more rational and consistent with other parameter paths.

- *variation* selects currently-displayed patch variation.
- *inst<layer#>/mute* replaces *unify/<layer#>/mute*.
- *inst<layer#>/solo* replaces *unify/<layer#>/solo*.
- *inst<layer#>/midiEnable* replaces *unify/<layer#>/midiEnable*.
- *inst<layer#>/midiToggle* replaces *unify/<layer#>/midiToggle*.

These new parameters also appear in parameter-path menus to make them easier to select. (Note the older top-level paths beginning with *unify/* still work, for compatibility with older patches).

All of these are binary parameters. Values 0.5 or greater are treated as “on”. *midiToggle* is a special case: values 0.5 or greater toggle the layer's MIDI enablement, lower values are simply ignored. This is for compatibility with external MIDI CC controls, which may send a high value when pressed and a low value when released: pressing triggers the action, release does nothing.

## Macro paths

Paths beginning with *macro* and followed by a natural number identify one of Unify's macro parameters.

- *macro/<macro#>*

Note these can't be used directly as links on macro knobs, i.e., a macro knob cannot link directly to itself or another macro knob in the *same Unify instance*.

They are mainly intended for use in MIDI Controller Files.

Macro knobs can, however, link to a macro control of an *embedded Unify instance*, by referring to the parameter-name (*Macro1*, *Macro2*, etc. or equivalent custom names in Unify v1.6.0 and later) in a *layer-specific path*; see below.

## Layer-specific paths

The majority of parameter paths used in Unify are *layer-specific*; they begin with one or more components which select a specific layer in the current patch (e.g. “inst/1”, “aux/2”, or “master”). Among these, only *master/volume* identifies a single parameter. All the others use additional components which further refine the selection of individual parameters specific to that layer.

- midi/<layer#>/[MIDI Layer params]
- inst/<layer#>/[INST Layer params]
- aux/<layer#>/[AUX Layer params]
- master/volume
- master/[AUX Layer params]

The MIDI, INST, and AUX/MASTER parameters use additional path components (sub-path), whose validity may depend on the type of layer, as follows:

sub-path	MIDI	INST	AUX	MASTER
midiEnable	x	x		
midiToggle	x	x		
mixLevel	x	x	x	
panPos	x	x		
directSend	x			
auxSend/<aux#>	x			
plugin/[Plug-in params]	x	x		
effect/<eff#>/[Plug-in params]		x	x	x
midiEffect/<eff#>/[Plug-in params]	x	x		

## Plug-in parameters

Individual plug-ins on all layers can be controlled by paths ending in “[Plug-in params]” in the table above. Starting with Unify v1.7, MIDI effect plug-ins must be referred to using “midiEffect”; in earlier versions “effect” could be used, but this is now reserved for audio-effect plug-ins only.

In general, plug-in parameters are the parameters which each individual plug-in exposes for host automation. Unify v1.7 introduced the concept of “special” plug-in parameters, which always begin with the underscore character (“\_”); the first of these is “\_bypass”, which allows controlling whether or not individual plug-ins are bypassed. Examples:

- *midi/1/midiEffect/2/\_bypass* bypasses the 2nd MIDI-effect plug-in on the MIDI1 layer
- *inst/2/midiEffect/1/\_bypass* bypasses the 1st MIDI-effect plug-in on the INST2 layer
- *inst/5/effect/3/\_bypass* bypasses the 3rd audio-effect plug-in on the INST5 layer

Whether setting the value of these *\_bypass* parameters via macro-knob links or by OSC, setting the parameter value to 0.5 or greater bypasses the plug-in; values below 0.5 un-bypass it.

# Patch Variations

This page describes an **advanced feature** which few Unify users will need, and which is still somewhat experimental at the time of writing.

To use the techniques described here, you should be comfortable with creating and editing **plain-text files**. However, the first example includes a link to a `.guru` file you can download and try, even before you understand all the details.

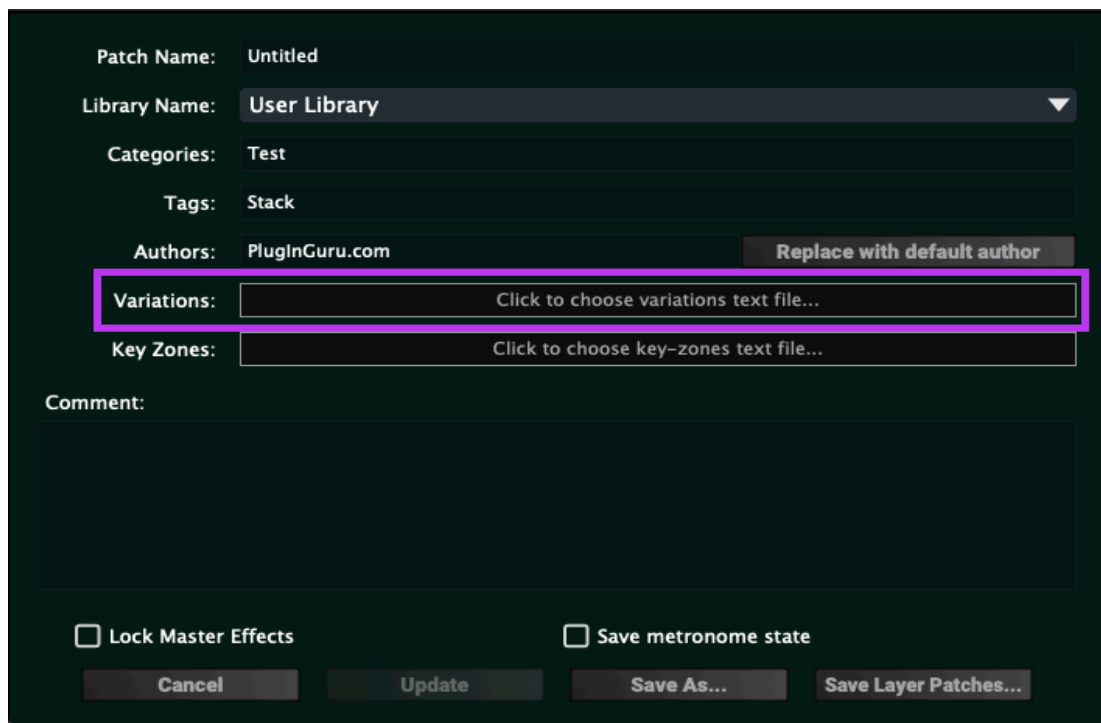
**Important Note:** when loading a patch from a library, Unify always expects to find the variations-list file in *that library's own Media folder*. This means, if you re-save a patch into a different library, you will need to copy the original variations-list file into that library's *Media* folder (and you may need to create the folder, if there isn't one already). **This is a limitation in Unify v1.6.x.** It may be fixed in a future update, but this is not guaranteed.

---

## Basic Concept

*Patch variations* were introduced in Unify v1.6.0. Any patch can optionally include a reference to a plain-text “variations” file (kept in the library's *Media* folder), which is simply a list of “variation names”, one per line. The newly-added **top-level parameter variation** can be linked to a macro knob, and simply selects which of the variation names is displayed in white to the right of the patch name.

You select the variations file for a patch (if any) when saving the patch. As of Unify v1.6.0, the patch-save dialog includes a “Variations” section (which was expanded in v1.8) as shown below:



To select (or change) the referenced variations file, click the *Variations* button to select the variations file you created from a pop-up menu:

- The first item allows you to *Clear* the variations file selection (if you decide you don't want to use one)
- After this, you will see a list of all *.txt* files in your library's *Media* folder, from which you can select.
- If you hold down Option/ALT when clicking the button, there will also be an *Open...* item near the top. This will pop up a standard file-open dialog, so you can navigate to any location to find the *.txt* file you want. *Note this will make your patch non-portable to other machines.*

## Example 1: Song sections

One very good use of patch variations is for patches containing one embedded-Unify INST layer for each part of a performance (e.g. Intro, Verse, Chorus), which are selectively enabled via *midiEnable* parameters linked to a macro knob. As the knob is turned, the layers are enabled one at a time, and the variation-text at the top of the GUI changes to match.

(Unify 1 screenshot for reference..same would appear in Unify 2)



## Variations file

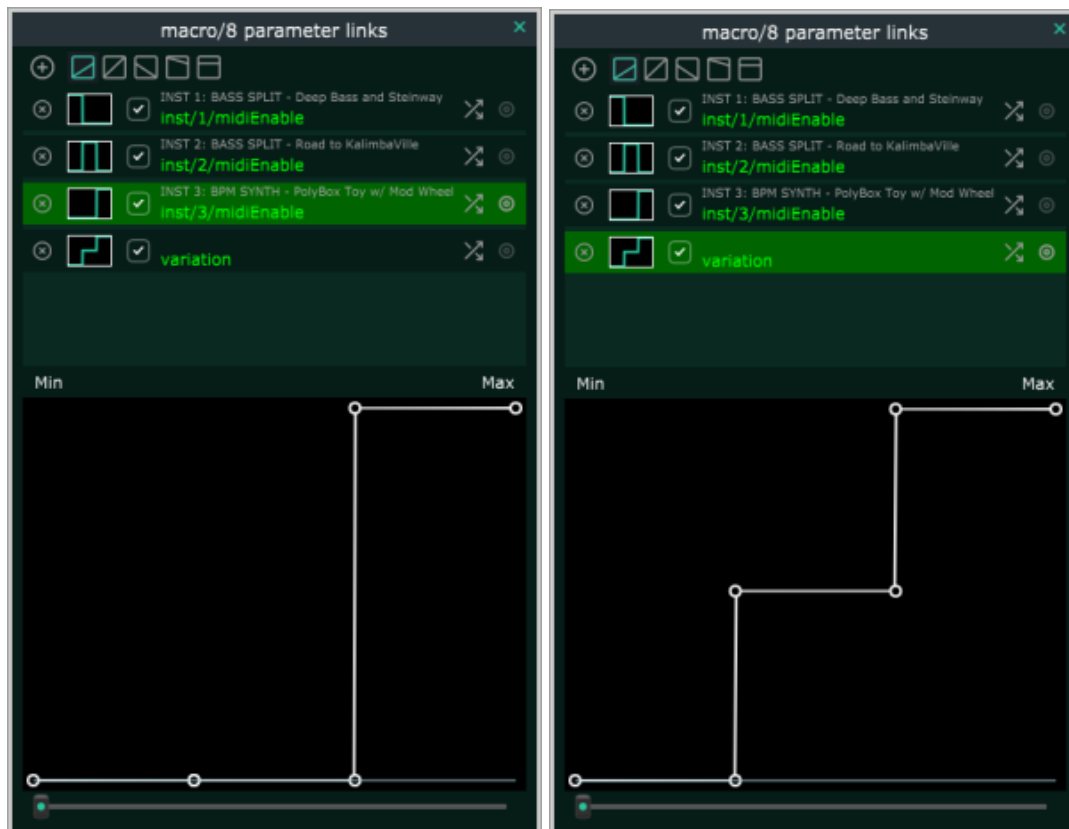
For this example, the variations-list file (*Intro\_Verse\_Chorus.txt*, found in the *Media* sub-folder of the *User Library* folder) is very simple, consisting of just three lines:

```
Intro  
Verse  
Chorus
```

## Parameter links

In the example *3 Parts Song Switcher* patch, macro knob #8 has been set up with some carefully-crafted links so that as the knob is turned, the layers are enabled one at a time, and the variation-text at the top of the GUI changes accordingly.

Here is what the links look like:



*Images from Unify V1 used for this example.*

## Example 2: Sample maps

Patch variations can also be used in combination with one or more sample-maps in Guru Sampler, where the parameter *osc1SampleSelect* is linked to the same macro-knob as *variation*, and the names in the variations file correspond one-to-one with the available choices on the sample-select menu in *Guru Sampler*. Provided the lines in the variations file correspond one-to-one with the *Guru Sampler* sample-select menu choices, and the *exact same response curves* are used in both parameter links, the variation-name displayed in large white text at the top of the Unify GUI will always indicate which sample is selected in *Guru Sampler*, so it's not necessary to have the sampler GUI open to see it. This technique was first used in the Unify library *PlugInGuru Chill GTR Phrases*.

### Variations file

Here is an early draft of a variations file from the *PlugInGuru Chill GTR Phrases* library. This file is called *Moody Chill GTR Phrases LIST.txt*, and is placed in the library's *Media* folder.

```
ALL Samples
01 Dm
02 Gmaj7 Gm/Bb
03 Bbm Ebm
04 Bbm
05 Em Fm Bm
06 Am D Em
07 Eb
08 Dm Am G
09 Dm Em Fm Am
10 Gm Eb Bb Cm
11 Am
12 A
13 Gm C5 Ab5/7
14 Dm
15 Bb C Dm
16 Gm F Eb
17 Fm Eb
18 E9sus4 G A
19 F Gm Cm Bb
20 Cm Bb Ab
21 Fm
22 C5 D5
23 Em Cmaj7
24 Dm
25 Dm Bb
26 Em Cmaj7
27 Csus2 Eb Csus2 Eb
28 Gm_Eb Gm_C Gm F
29 Am Em Am Em
```

```

30 Gm Eb Cm
31 Dm Am G
32 Am F C B
33 Dm C G
34 Em C G D/F
35 Em C
36 D A6 Fm B5
37 Gm Dm F
38 Bbm Dbsus2
39 Gm Bb add9
40 F G Am C Dm
41 Cm Ab Cm Eb
42 Bb F_A
43 Gm Eb Gm Eb
44 Cm Bb Gm Ab
45 Cm Bb Gm Ab
46 Gm
47 F C Am
48 G D C
49 Gm D5
50 Cm Eb_Bb
51 Cm Ebmaj7
52 G5 Bb C5
53 G/B C Dsus4 G
54 Gm
55 Cmaj7 D
56 D5 C5 Bb
57 Cm Bb
58 Cm Bb
59 Am C G F
60 Am F
61 C5 Eb Fadd9

```

The initial “All Samples” line is not actually used in this library, but it must be present in this file AND the corresponding sample-names list file (see below), because *Guru Sampler* always provides a default “all samples” option as the first choice on the sample-select menu.

## Sample-names list files

*Guru Sampler* supports optional *Sample-names list files* sibling to *.gsd* or *.sfz* sample-map files (with the same file name, but extension *.txt*). These sample-names lists must correspond one-to-one with the available samples in the sample-map, plus an extra initial “all samples” entry.

Here is an example from the *PlugInGuru Chill GTR Phrases* library, a file called *120 Moody Chill GTR Phrases.txt* (there are several, and their contents are identical) in the library's *Samples* folder.

Guitar Phrase #01 - Dm  
Guitar Phrase #02 - Gmaj7 - Gm\_Bb  
Guitar Phrase #03 - Bbm- Ebm  
Guitar Phrase #04 - Bbm  
Guitar Phrase #05 - Em - F#m - Bm  
Guitar Phrase #06 - An - D - Em  
Guitar Phrase #07 - Eb  
Guitar Phrase #08 - Dm - Am - G  
Guitar Phrase #09 - Dm - Em - Fm - Am  
Guitar Phrase #10 - Gm - Eb - Bb - Cm  
Guitar Phrase #11 - Am - Am - Am - G\_C  
Guitar Phrase #12 - A  
Guitar Phrase #13 - Gm - C5 - Ab no3 Maj7  
Guitar Phrase #14 - Dm  
Guitar Phrase #15 - Bb - C - Dm  
Guitar Phrase #16 - Gm - F - Eb  
Guitar Phrase #17 - Fm - Eb  
Guitar Phrase #18 - E9 sus4 - G - A  
Guitar Phrase #19 - F - Gm - Cm Bb  
Guitar Phrase #20 - Cm - Bb - Ab  
Guitar Phrase #21 - Fm  
Guitar Phrase #22 - C5 - D5  
Guitar Phrase #23 - Em - Cmaj7  
Guitar Phrase #24 - Dm  
Guitar Phrase #25 - Dm - Bb  
Guitar Phrase #26 - Em - Cmaj7  
Guitar Phrase #27 - Csus2 - Eb - Csus2 - Eb  
Guitar Phrase #28 - Gm\_Eb - Gm\_C - Gm - F  
Guitar Phrase #29 - Am - Em - Am - Em  
Guitar Phrase #30 - Gm - Eb - Cm  
Guitar Phrase #31 - Dm - Am - G  
Guitar Phrase #32 - Am-F-C -B  
Guitar Phrase #33 - Dm - C - G  
Guitar Phrase #34 - Em - C  
Guitar Phrase #35 - Em - C  
Guitar Phrase #36 - F#min  
Guitar Phrase #37 - Gm -Dm - F  
Guitar Phrase #38 - Bbm - Db sus2  
Guitar Phrase #39 - Gm - Bb add9  
Guitar Phrase #40 - F - G - Am - C - Dm  
Guitar Phrase #41 - Cm - Ab - Cm - Eb  
Guitar Phrase #42 - Bb - F\_A  
Guitar Phrase #43 - Gm - Eb - Gm - Eb  
Guitar Phrase #44 - Cmin - Bb - Gm - Ab  
Guitar Phrase #45 - Cmin - Bb - Gm - Ab  
Guitar Phrase #46 - Gm  
Guitar Phrase #47 - F - C - Am

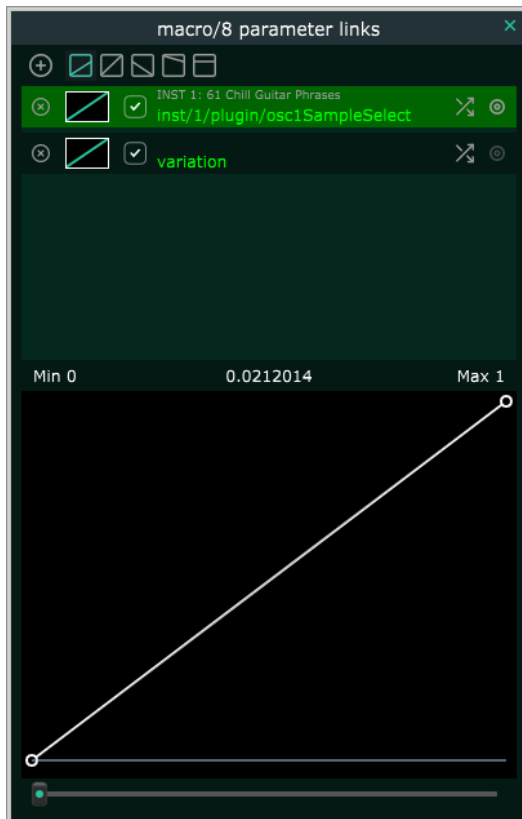
Guitar Phrase #48 - G - D - C  
Guitar Phrase #49 - Gm - D5  
Guitar Phrase #50 - Cm - Eb\_Bb  
Guitar Phrase #51 - Cm - Eb maj7  
Guitar Phrase #52 - G5 - Bb - C5  
Guitar Phrase #53 - G\_B - C - D sus4 - G  
Guitar Phrase #54 - Gmin  
Guitar Phrase #55 - Cmaj7 - D  
Guitar Phrase #56 - D5 - C5 - Bb  
Guitar Phrase #57 - Cm - Bb  
Guitar Phrase #58 - Cm - Bb  
Guitar Phrase #59 -Am - C - G - F  
Guitar Phrase #60 - Am - F  
Guitar Phrase #61 - C5 - Eb - F add9

Observe how these sample-names correspond one-to-one, to the names in the variations file, *except there is no default "all samples" line at the start*. This is because this is a list of *sample names*, not *mapping options*.

The exact text in the two files need not be the same; library author John Lehmkuhl has chosen the text to work well in the large variation-name display in the former case, vs. the small sample-select menu in the latter.

## Parameter links

Here is an example of how one of the Unify macro knobs is linked to both the *osc1SampleSelect* parameter of a *Guru Sampler* instance, and the top-level (global) *variation* parameter, in the *PlugInGuru Chill GTR Phrases* library:



*Image from Unify V1 used for this example.*

What's not immediately obvious in this screenshot is that the leftmost end of the response curve *does not go all the way down to zero*, because that would select the default "all samples" map, which is not used in this case. Instead, the leftmost point has been carefully positioned to select the *second* choice, which in this case is the single sample "Guitar Phrase #01 - Dm". Then the curve is copy/pasted to the second link to *variation*.

## Important reminders

- Always include an initial entry in the variations file, corresponding to the default “all samples” choice.
- The sample-name list files used with *Guru Sampler* do *not* include the “all samples” item.
- Parameter response curves should normally be linear; there is little practical value in using curved shapes.
- If you don't want to ever select the “all samples” choice, make sure the lowest value of your response curve does not go all the way down to zero.

## About .guru files

.guru files provide a convenient way to package, distribute, and install batches of related files in Unify, e.g., patch libraries, sample collections, etc.

A .guru file is simply a renamed .zip file, whose contents (files and folders) are set up to unzip into Unify's main data folder. Getting the folder structure right is the trickiest part of making your own .guru files.

## Example .guru file

The following example .guru file was created in support of a new video about the [Patch Variations](#) feature in Unify v1.6.0:

[unify\\_step\\_curves.zip](#)

Click the link above to download this .guru file, and note that *it has been re-zipped*, i.e., you will have to unzip it after downloading. The software powering this online manual does not support the .guru file type, so it was necessary to zip it again before uploading. This is a common limitation of online file-hosting tools.

This particular .guru file has the following folder structure:

- file: *Unify Step Curves.guru*
  - folder: *Libraries*
    - folder: *User Library*
      - folder: *Media*
        - file: *Intro\_Verse\_Chorus.txt*
      - folder: *Patches*
        - file: *3 Parts Song Switcher.unify*
    - folder: *Presets*
      - folder: *Macro Link Curves*
        - 18 .curve files

When this .guru file is dropped into the Unify GUI, Unify will unzip its contents into the main data folder. Hence, everything inside the *Libraries* folder in the .guru file will go into the *Libraries* folder in the main Unify data folder, and everything inside the *Presets* folder will go into *Presets*.

What happens is exactly the same as if you put the .guru file itself into the main Unify data folder, and unzipped its contents using the “extract here” option. (Different systems may use different wording to describe this, but the effect is the same, as described in the next section.)

## How Unify unzips the .guru file

Any folders present in the .guru file, which don't yet exist in the corresponding locations under the main Unify data folder, will get created automatically. This is why it is VERY important to get the folders inside your .guru files just right, or files and folders will go into the wrong places, and anyone to whom you send your .guru file will end up with a mess to clean up.

Individual Files present in the .guru file will be created if they don't already exist, but if a file by the same name does exist in that location, it will be overwritten with the new file. This is another reason to be very careful when setting up .guru files.

If the .guru file contains any patch files (i.e., files which go into the Patches sub-folder of any Library folder), Unify will take note of the library name (i.e., the name of the parent folder of the Patches folder containing the patch file) and trigger the patch database to rebuild only for that library (or libraries, if more than one are present).

## Limitations of .guru files

When Unify unzips the contents of a .guru file, it can only *add* or *overwrite* files (and folders as required). *It cannot delete files.* Some capability to do this may be added to Unify in the future, but there is none at the moment.

## Creating a .guru file (the safe way)

Suppose you have created a new library called *My Awesome Library* in Unify, and you'd like to make a .guru file to help others install it quickly. Here are the steps:

1. Start by creating an empty “working folder” somewhere, preferably *not* under the main Unify data folder.
2. Open your working folder, and also the main Unify data folder, in *Windows Explorer* or *Mac Finder*.
  - To quickly open the main Unify data folder, click the “gear” icon to go to the **Settings view**, and click the “Open...” button there.
3. In the window representing your working folder:
  - Create an empty *Libraries* folder, and open it
4. In the main Unify data folder window:
  - Navigate into the *Libraries* folder
  - Locate the folder for your library (in this case *My Awesome Library*)
5. **Copy** the entire *My Awesome Library* folder to the empty *Libraries* sub-folder of your working folder
6. In your working-folder window, navigate back up to where you started (where you see the *Libraries* folder you made at step 3).
7. Select the *Libraries* folder, and zip it, using whatever mechanism your computer provides to make .zip files, e.g.:
  - On a Mac, right-click the *Libraries* folder and choose “compress”
  - On a Windows PC with 7-zip installed, right-click the *Libraries* folder and choose “add to ... .zip” (the name will be the name of your working folder).

8. Rename the resulting `.zip` file to whatever you want, and *change the extension from `.zip` to `.guru`*.
  - Your operating system will pop up some kind of message asking you to confirm that you really want to change the extension to `.guru`.
  - Click “OK” or respond in whatever is the appropriate way, to confirm that you want to change the extension.

At this point, assuming you followed the above steps precisely, you will have a valid `.guru` file, which others can now use to install *My Awesome Library* quickly.

If you want to upload your `.guru` file to some kind of file-sharing service, attach to an email, etc., it's usually best to *re-zip* the `.guru` file, and share/attach the resulting `.zip` file. Advise recipients that they must unzip it, then drag/drop the resulting `.guru` file into the Unify GUI, or use the *Select .guru file...* button in the Settings view and select it that way.

## Creating a `.guru` file (the risky way)

If you know what you're doing, you can use these steps instead:

1. Open the main Unify data folder, and navigate into *Libraries*
2. Create a new empty *Libraries* folder (inside the *Libraries* folder)
3. Drag (i.e., move) your own library folder into this new *Libraries* folder
4. Compress the new *Libraries* folder to a `.zip` file (e.g. *Libraries.zip*), then rename using the `.guru` extension, and confirm as described above.
5. Make sure to move your library folder back out into the main *Libraries* folder, then delete the dummy *Libraries* sub-folder you created.

## Creating a `.guru` file (more generic)

The above instructions described how to make a `.guru` file for a single patch library, but `.guru` files can contain multiple libraries, or no libraries (i.e., no top-level *Libraries* folder at all), and anything else that belongs under the main Unify data folder, e.g., you may have a top-level *Presets* folder as in the example, or *MIDI Bank Files*, *MIDI Controller Files*, etc. In such cases, it's strongly advisable to use a working folder.

Whatever you do, just make sure that the folder structure of your `.guru` file exactly corresponds to where all of your own files need to go.

## Testing your `.guru` files

It's a very good idea to test dragging your `.guru` file into Unify, to make sure everything goes into the right places. The best way to do this is to create a second “main Unify data folder” somewhere on your disk, and select in Unify using the “Change...” button in the Settings view. That way, there will be no risk of messing up your actual Unify data folder contents, and you can switch back after you're finished testing.

To create a second Unify data folder:

1. Create an empty folder in a suitable location on your disk, then create empty *Libraries* and *Presets* folders inside it.
  - If you do not create these sub-folders, Unify will not allow you to select the folder as your main Unify data folder.
2. In Unify, go to the Settings view, click the “Change...” button, and choose your new folder.
3. **IMPORTANT: Quit Unify** immediately.
  - This step ensures that your patch database will not be damaged.
4. Optionally, copy over any other content you may want (patch libraries, presets, etc.) from your real Unify data folder to your testing folder.
  - Be very careful that to COPY (not MOVE) files/folders to your test data folder
5. Re-start Unify
  - Unify will automatically create an empty database file *presets.db* in your test *Libraries* folder.
  - If you copied over any libraries, click the lightning-bolt icon at the top of the patch browser, to update this newly-created database to include entries for whatever patches you copied.
6. Optional, but recommended: Quit Unify again, and zip up your entire test folder.
  - This will allow you to delete and re-create it quickly, for future tests.
7. Drag/drop your *.guru* file, and verify that Unify responds as you would wish.
8. Examine the contents of your test data folder carefully, and verify that all files/folders went into the right places.

If you notice any problems at steps 7/8, you will need to re-build your *.guru* file accordingly. Before trying it again, delete your entire test data folder, and re-create it as it was at step 6, by unzipping the zip file you created at step 6. Then you can simply restart Unify and repeat steps 7/8.

When you are finished testing:

1. Go to Settings, click the “Change...” button, and re-select your original Unify data folder.
2. **Quit Unify** immediately, then re-start
  - Everything should now be back exactly as it was, including your patch database.

## Keeping Unify's patch database clean with scrub.xml

Part-way through 2020, we realized that a large number of presets in the Standard Library had inconsistent capitalization. We also realized that it's so easy to make simple errors when entering Tag/Category text for new patches, that end-users would likely make the same kinds of mistakes all the time.

To address this problem without having to actually update hundreds of Unify patch files, we devised a simple system whereby Unify actually *rewrites* these patch-metadata items as it rebuilds the patch database. These rules live in a small text file called *scrub.xml* and are described below.

What must be emphasized is that Unify applies these rules *when writing to the patch database*; it does NOT change anything in the patch files themselves.

This means that if you don't like the result, you can simply edit the *scrub.xml* file (to change or delete some of the rules) and rebuild the database.

### Location and format of the scrub.xml file

The *scrub.xml* file lives in Unify's main *Libraries* folder, which itself is in your main Unify data folder. To locate it easily, go to Unify's Settings view and click the “Open...” button. This will open the main Unify data folder in the Finder (Mac) or Explorer (Windows). Locate the *Libraries* sub-folder and double-click to open it. You'll see more folders, one for each Unify library, and a few other files, one of which is *scrub.xml*.

***scrub.xml* is a “plain-text file”** and MUST be opened using a true text-editor app such as *Notepad* on Windows (or any “programmer's editor” app such as Notepad++, Sublime Text, etc.). On a Mac, you can use *TextEdit*, but you MUST make sure you have it set to *plain text mode*.

***scrub.xml* is an XML file.** Please DO NOT attempt to edit it unless/until you are familiar with the XML file format.

## Structure of the scrub.xml file and rules

Here is the exact contents of the *scrub.xml* file distributed with Unify v1.3.2:

### Scrub.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<Rules>
  <Fixed word="Arp"/>
  <Fixed word="BPM"/>
  <Fixed word="DDL"/>
  <Fixed word="FM"/>
  <Fixed word="FX"/>
  <Fixed word="LFO"/>
  <Fixed word="MegaMagic"/>
  <Fixed word="RX2"/>
  <Fixed word="PD"/>
  <Fixed word="SuperMacho"/>

  <Rewrite word="Drums" as="Drum"/>
  <Rewrite word="Hihats" as="Hihat"/>
  <Rewrite word="Chords." as="Chord"/>
  <Rewrite word="Chords" as="Chord"/>
  <Rewrite word="Hits" as="Hit"/>

  <Field name="tags" fixed="1" rewrite="1"/>
  <Field name="category" fixed="1" rewrite="1"/>
</Rules>
```

The top-level XML tag is *Rules*, and inside it are a series of child tags representing individual rewrite rules, as follows:

- `<Fixed>` specifies that a particular word is always to be rewritten with the specified capitalization.
- `<Rewrite>` specifies that a particular word is always to be rewritten exactly as specified.
- `<Field>` indicates that a named field in the “Presets” table is subject to the `<Fixed>` and/or `<Rewrite>` rules.

# Unify's files and folders

This page sheds some light on a few key aspects of how Unify works with files and folders on your computer. Although we have done our best to make Unify install and work automatically, there will be times when you'll need to understand where Unify's various files live, and what they are there for.

## Four distinct groups of files

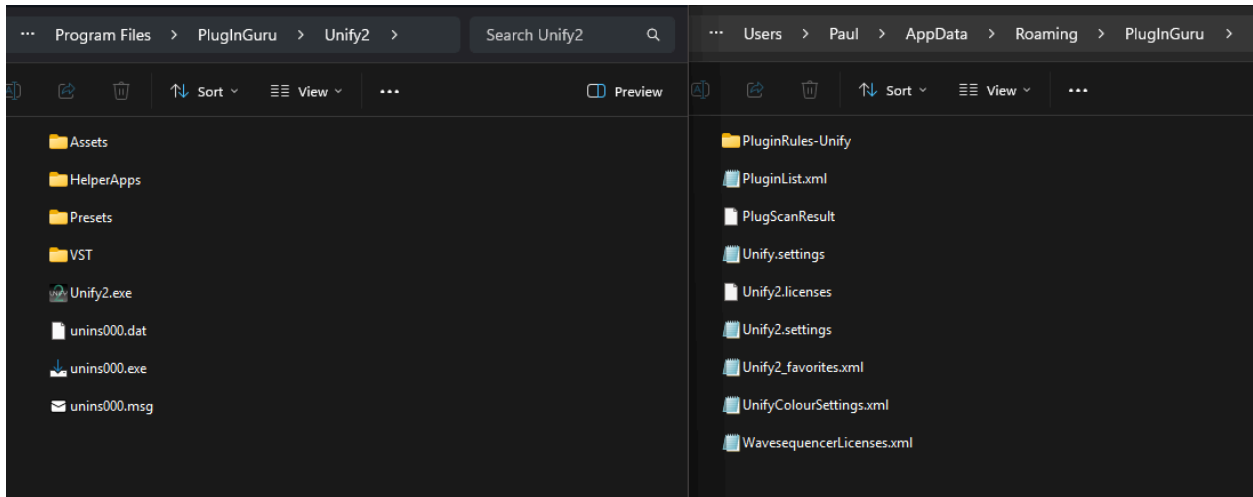
Unify uses four groups of files, which are stored in three very different places. The exact places are different for Windows PCs and Macs; see the next two sections for the details

1. Unify's **app folder** contains the Unify stand-alone app, the bundled third-party VST plug-ins, and a “helper” app for scanning your own plug-ins.
2. The **Unify plug-ins** (the ones you load into a DAW) are stored together with your other plug-ins.
3. Unify's **content folder** is a single folder containing *libraries*, *presets*, and *patch database*.
4. Unify's **settings** (preferences) and its **plug-in database** are kept together in a special folder

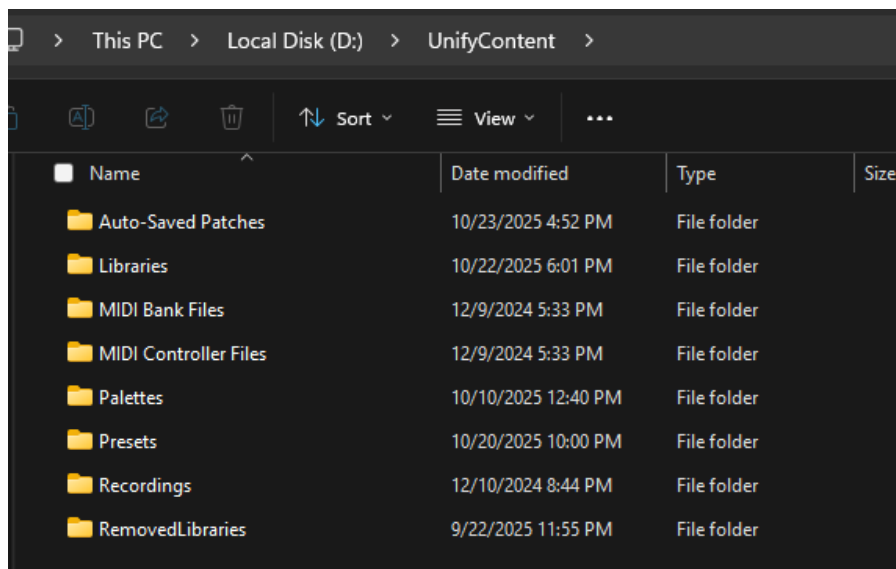
## Finding Unify's files on a Windows PC

1. **App folder:**
  - *C:\Program Files\PluginGuru\Unify 2\*
  - This location is fixed by the Unify installer, and should not be changed.
2. **Unify plug-in:**
  - VST3: *C:\Program Files\Common Files\VST3* (this is the standard VST3 location for Windows).
3. **Content folder:**
  - Wherever you chose to put this, when you ran Unify for the first time
  - Default location is *C:\Users\Public\Documents\PluginGuru\Unify*
  - Current location is displayed in Unify's Settings view, where there is also a button to change it
  - (Note: You must move or copy the content manually before changing; Unify does not do this automatically.)
4. **Settings:**
  - On Windows 7 and later, there is a hidden *AppData* folder inside *C:\Users\your username\*. Unify's settings folder lives in the *Roaming* sub-folder, and is called *PluginGuru*.
  - A useful reference is <https://support.pointlogic.com/faq/troubleshooting/accessing-the-appdata-folder>
  - A quick way to get to *C:\Users\your username\AppData\Roaming\* is to type `%appdata%` into the title bar of any Windows Explorer window and press
  - ENTER.

Unify 2 App and supporting files / App Data folder (Fixed locations set by the installer)



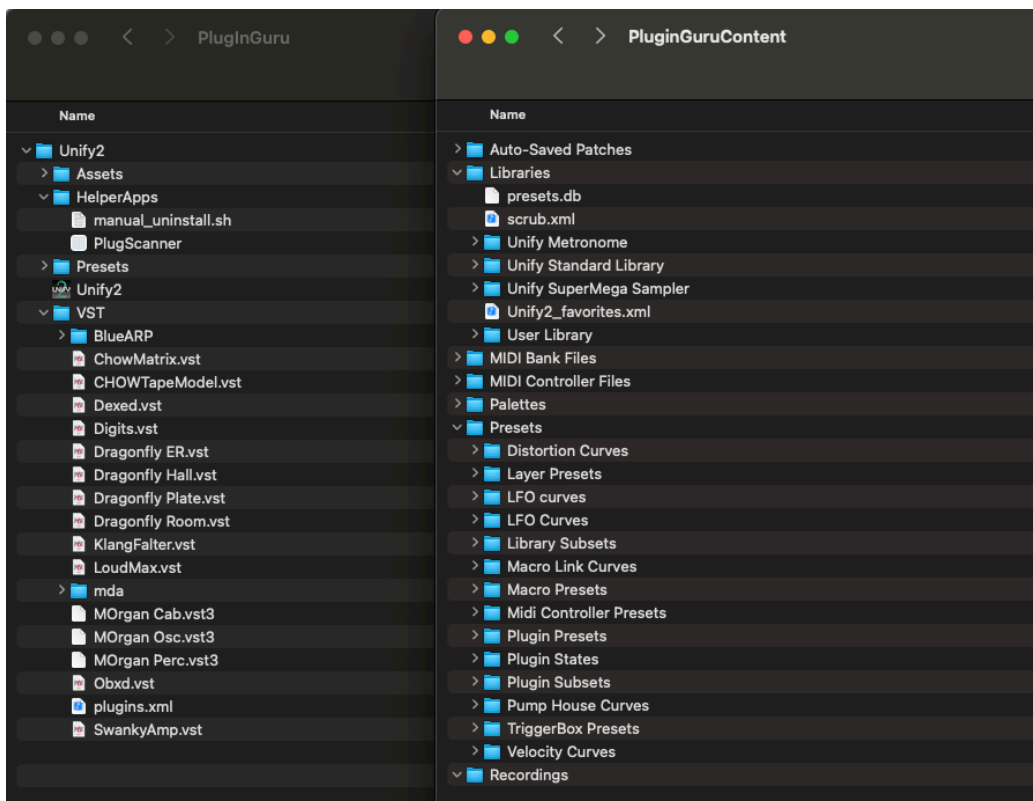
Unify 2 content (user selectable location)



Note you may see more or fewer files/folders than indicated in the above screenshots.

## Finding Unify's files on a Macintosh

- 1. App folder:**
  - The Unify stand-alone app lives in `/Applications/PlugInGuru/Unify 2`
  - Bundled VSTs and “helper” apps live in `/Applications/PlugInGuru/Unify 2`
  - These locations are set by the Unify installer, and should not be changed.
- 2. Unify plug-ins:**
  - Audio Unit: `/Library/Audio/Plug-Ins/Components/Unify.component`
  - VST3: `/Library/Audio/Plug-Ins/VST3/Unify.vst3`
  - These are the standard Mac plug-in locations, and should not be changed.
- 3. Content folder:**
  - Wherever you chose to put this, when you ran Unify for the first time
  - Default location is `/Users/Shared/PlugInGuru/Unify`
  - Current location is displayed in Unify's Settings view, where there is also a button to change it
  - (Note: You must move or copy the content manually before changing; Unify does not do this automatically.)
- 4. Settings:**
  - `/Users/your username/Library/Application Support/PlugInGuru`



(Application folder (fixed location - set by the installer) / Content folder (user choice of location))

Note you may see more or fewer files/folders than indicated in the above screenshots.

## Troubleshooting basic file/folder issues

### Leftover .guru files

Unify libraries (and certain other files under the content folder) are delivered in the form of *.guru files*. These are basically *.zip* files, and they contain a folder structure which mimics that of Unify's main content folder. When Unify opens a *.guru* file (either because you drag it into the Unify GUI, or use the "Select .guru file..." button in Settings view), it unpacks the zipped files into the appropriate content folders.

We use *.guru* files to install and update Unify content, instead of a regular installer, for two reasons:

1. Only Unify "knows" where you have chosen to keep the content folder (stored in the *Unify.settings* file); installers do not.
2. Installers are mainly for executable files like programs and plug-ins, and are treated specially by the operating system to facilitate things like un-installing. Unify's content files are just ordinary data files, which don't require such special treatment.

**Once Unify has unpacked a .guru file, you can discard the file.** The contents have been copied into the proper places in the Unify content folder, so the *.guru* file is no longer needed, and is just taking up valuable disk space. You may wish to keep *.guru* files for backup, but this is not strictly necessary, because you can always re-download them later.

## Unify 2 vs Unify 1 - what's new?

### Layers:

The layers mode features the original layers view of each of the MIDI, instrument, aux and master layers, however the aux and master sections are now permanently visible at the bottom half of the view.

The layers can be individually minimized or maximized by clicking the chevron button at the right hand side of each layer - right clicking will minimize or maximize the whole group of related layers.

In the minimized mode you have access to a reduced set of controls, but on instrument layers the central area can be switched via the burger menu - you can switch between the key-zones editor, the MIDI box, the instrument box, the audio effects box and the aux sends box.

Right clicking and selecting the option will apply it to all instrument layers.

In the expanded mode for instrument, aux and master layers there is a dedicated row showing all the plugin boxes. When there isn't enough horizontal room for plugins, there will be a right/left chevron button you can click to scroll through the loaded plugins.

MIDI and instrument layers now get a more visual MIDI zones editor where you can clearly see incoming note data and adjust the note ranges.

In expanded mode, we also now show the instrument audio load graph at the top of the central area of the layer.

The idea of the minimized mode is to give a more simple overview of lots of layers - which is more useful for quick patch browsing and level balancing, without overwhelming you with options.. and then you can expand the view to dig in and tweak more. It also lets more of the background image show through in most patches.. more on that later.

### Filter buttons:

New to Unify 2 are the always accessible MIDI filter shortcut buttons - these allow you to disable or enable pitch bend, mod wheel, aftertouch, expression pedal, sustain per layer - this can be really handy for example if you want pads on one layer and to play a lead in the other layer - so you could for example disable pitch bend on the pads layer and disable sustain on the lead layer.

The last button in that area is the latch button, which becomes enabled if the MIDI source/input is set to Latch mode. When you enable audio inputs or multiple audio outputs (either via the DAW settings or hardware interface settings in standalone mode) you will see a selector on the left and right hand side of instrument and aux layers.

These let you pick a hardware input or output to send audio to.

The default plugin on instrument layers is the 'sine wave synth' - if it's loaded, selecting an audio input channel will automatically cause the audio input plugin to get loaded into the layer.

MIDI out:

MIDI out button - you can enable MIDI out on MIDI and instrument layers - the MIDI icon can be toggled on or off.. it's used to send MIDI to the main MIDI hardware output (in standalone mode), or you can forward MIDI data from Unify to your other DAW tracks. This is not supported in Logic because AU plugins don't support that feature.

Play mode:

The play mode presents a simple view with the patch name and a selection of macro knobs/controls that have been enabled in the patch.

There is a virtual keyboard which can be optionally hidden, and there is a picture button which allows selection of a background image from a built in library of images. These images are known as the 'embedded' images..

When loading patches, the default option will be to load any image associated with the patch name or library within the associated media folder - if one has been specified.

If you save a patch without defining a custom image in the media folder links text file, it will be saved with the currently selected embedded image - which will be loaded on next patch load.

You can pin the background image so that it remains selected even when changing patches.

The tool button switches the panel into edit mode, where patch developers can select which macros are visible, select the layout, and select custom macro control colors.

Browser:

The browser functionality remains largely un-changed - however there are new buttons to set the loading mode, and the browser will switch to a double column mode if you stretch the UI wide enough.

The look of the browser is updated to make it more 'pretty' - there may be more options in future to select different backgrounds and more advanced browser views.

## Bottom bar/transport area/recording/metronome:

New to Unify 2, the transport related controls are always visible and give you a live view on the current playback time.

The trigger modes and host follow mode are always visible.

New features in Unify 2 on this bottom bar area are the ability to open the virtual keyboard and macro controls panel at the same time, the recording function, and the metronome function.

## The metronome:

The metronome is a parallel instance of Unify that can be opened just like an embedded instance of Unify; however it's designed to respond to a single input trigger note that is generated when the transport starts, and released when the transport stops.

This metronome layer can be used to drive drum/backing layers to which you can jam along with in the main instance.. or you could use it to drive a step sequencer or other MIDI effects ; there are many possibilities.

Metronome options are available via the panel opened with the chevron button above the metronome icon.

Right click the metronome to open the metronome Unify instance, or access it via the metronome options panel button.

## Recording:

The recording feature is a simple direct to disc recording feature that just starts recording MIDI and audio and saves it to the file system - you can access the latest recordings by right clicking on the record button after recording is completed, or by opening the panel (chevron above the record button) and click the files button.

Recording can be optionally set to only start after the first MIDI event using the MIDI trigger option. In the recording options panel, you can choose to record master, MIDI, layers and metronome in various combinations.

It can be useful to use the metronome layer to play along to a groove, but maybe you don't want to record those rhythm layers. Note that by default, in DAW hosted mode, the metronome audio will get mixed with the main master output audio.

In standalone mode, the metronome audio is mixed to the main hardware output along with the top Unify instance master bus audio - there is an option to insert metronome audio directly into the main instance master effect bus.

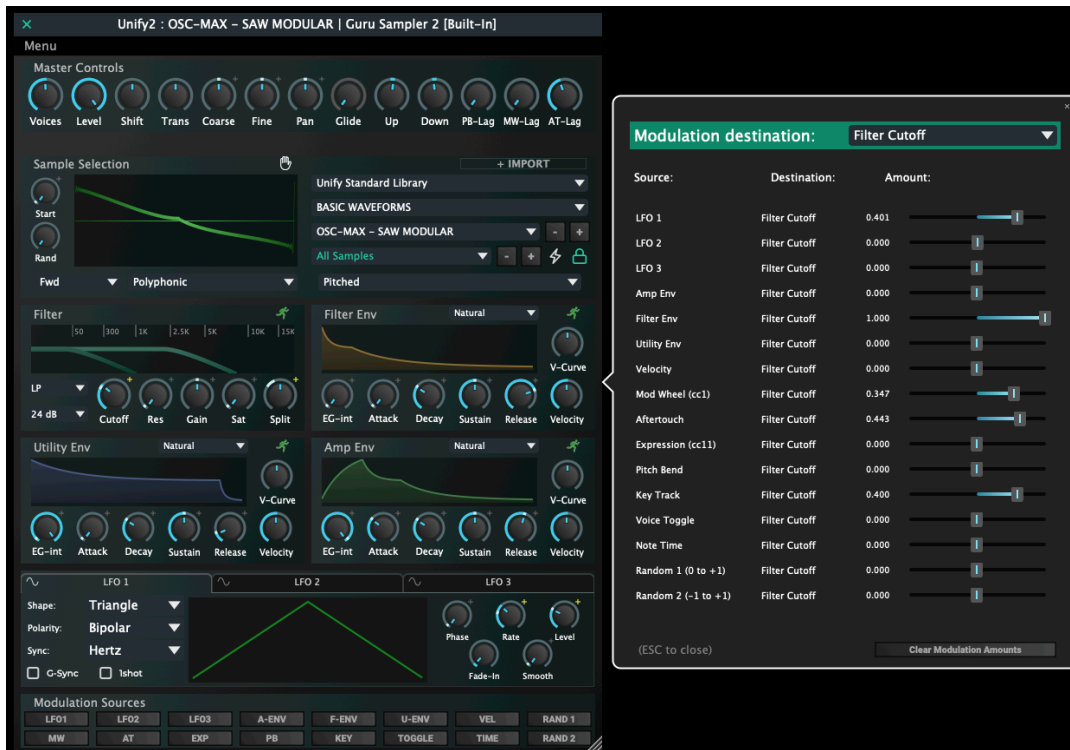
## New features for the macro controls:



Macro controls now feature 7 custom shapes a built in “LFO” that can be set to run or stop on any of the controls - there's a 'arm/play' button on each macro which can be used to either start/stop the LFO or arm it so that it can be controlled with the master Macro LFOs play button found at the far Right of the 8 Macro Knobs and on the Play Page next to the Keyboard Show On/Off button.

## New plugins (GuruSampler2 +)

### Gurusampler 2:



Unify 2 comes with a completely reworked version of GuruSampler, with a modern interface and many more features. GuruSampler 2 (GS2) now features: Animated/interactive visuals for samples, filter, envelopes and LFOs as well as visual animation of modulations.

A full mod matrix implementation for polyphonic modulation of each of the up to 64 enabled voices. Support for polyphonic aftertouch control. 3 types of envelope curve.

A 3rd utility envelope and 3 LFOs instead of one. Single sample import and auto creation of SFZ files available now (one sample mapped across the whole keyboard - a more advanced multi-sample creator/importer might come in future).

Filter enhancements:

Enhanced filter with low frequency gain compensation and auto high pass filtering at high resonances and low cutoffs to avoid 'blowing up' - which also enables more resonance to be applied.

Built in DC filtering to avoid DC offsets from 'bad' samples or due to setting the start point in short waveform. In filter saturation (pre-filtering). Split control for modulating/balancing the cut-off frequency on the left and right channels.

Modulation:

Multiple new modulation source types including expression, pitch bend, keytoggle, time (time between notes), random (unipolar), random (bipolar).

Lag control for aftertouch input.

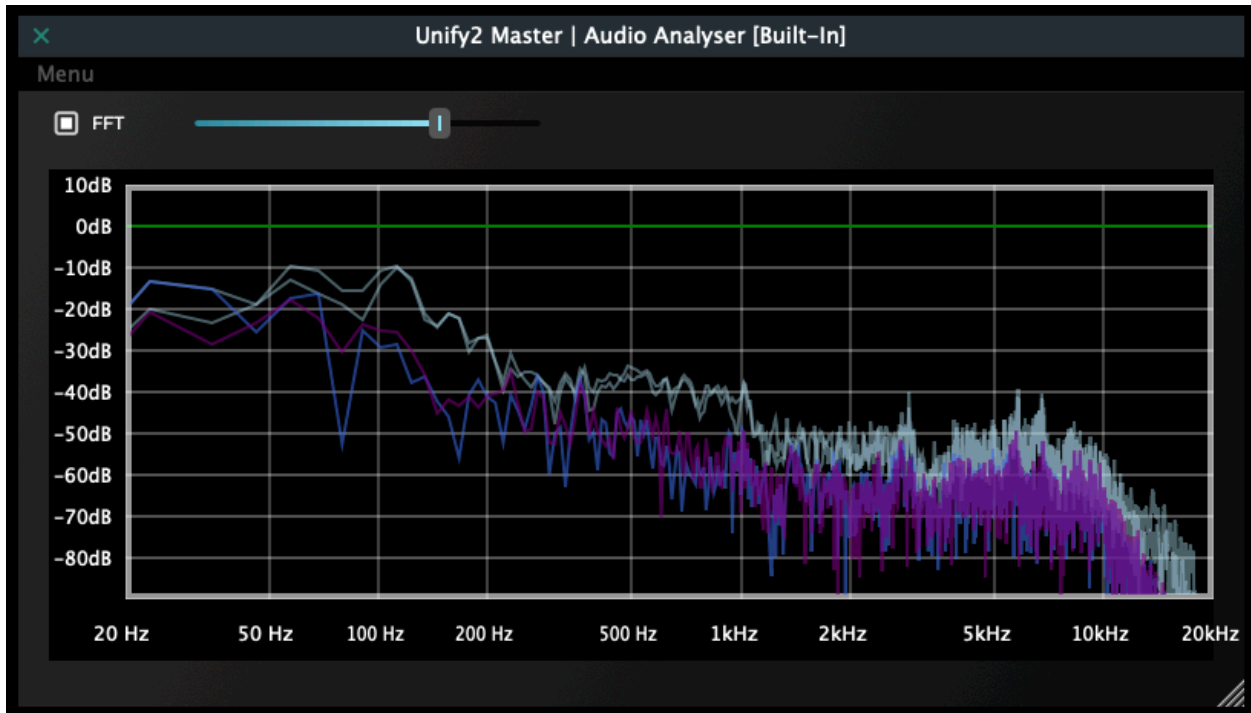
LFOs are polyphonic by default and now offer a fade-in and smoothing control - which can be useful for gradual vibrato or for smoothing out sharp transitions (the smoothing amount is also a modulation destination).

The sample start position is now a modulation target.

You can set modulation amounts directly on knobs (click the source button first), or via the mixer panel that is opened by right clicking on valid modulation target knobs.

GuruSampler 2 will demand a little more from your CPU (typically 25% to 50% more vs GS1) due to all the extra modulation going on, but it's optimized to only process modulations that are actually assigned.

## New audio analyzer plugin



Spectrum analyzer and oscilloscope.. this can be pretty handy to compare layers for example.. as you can put it on any layer (be aware there is CPU load cost using the spectrum analyzer).

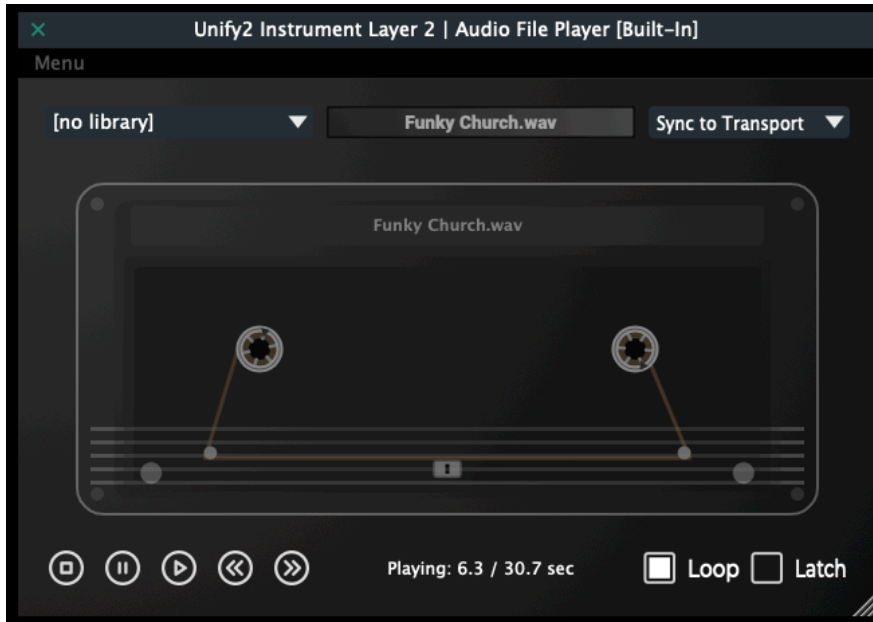
## New MIDI step sequencer plugin



Up to 64 steps with gate, pitch, velocity, random jumps, and lock to scale options.

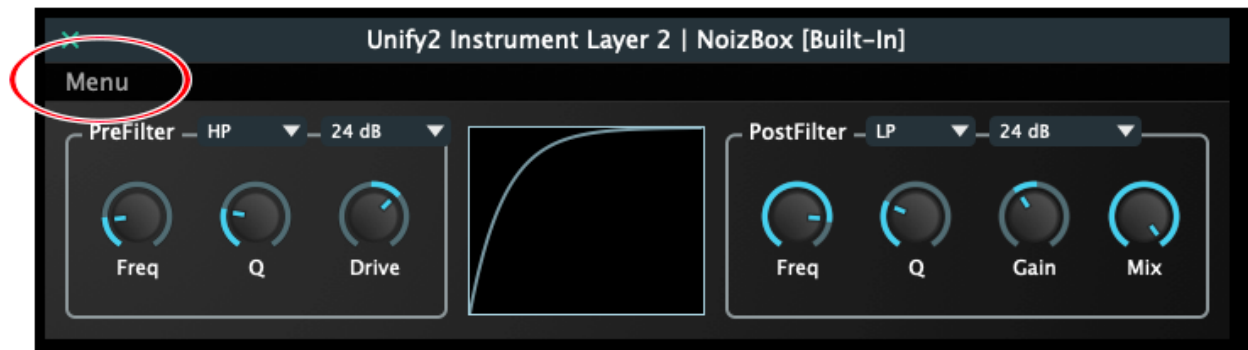
The step sequencer could be used to drive other layers which might also include other sequencers. You can use it with really long step times by using the bars range.. so for example it could drive a root/bass note, and then that could drive other sequencers playing a faster sequence based on the root note changes. It's also fun to use with external hardware.

## Updated audio file player plugin



The audio file player plugin has a new latch and loop option and an animated retro style cassette tape visual showing progress through the audio. Latch and loop can be really useful for long environmental background sounds for example.

## Plugin window features:



All the plugins hosted in Unify 2 now feature a menu bar at the top of the plugin window where you can save and load presets/states of the plugin, and save some 'template' presets - these will then appear within the menu of the plugin.. this could be handy for setting up some started patches for GuruSampler2 or maybe saving your favorite EQ or reverb settings for quick recall.