Welcome to Unify!

Hello and thank you for purchasing **Unify**! You navigate around these pages by clicking colored links, just like any other web site. Most pages have a table of contents box in the upper-right corner, containing links you can click on to quickly reach different sections on that page. If you ever get lost, just click on the “UN” Unify logo which appears at the top-left corner of every page, to get right back to this start page.

This Unify manual site is **under construction**, and probably will be forever, as the software itself changes over time. RED LINKS like this indicate pages which aren't available yet, because we're still working on them. Please excuse us when things are outdated or missing. If you think there's something really important we need to know about, head to the PlugInGuru forums to let us know. Thank you!

Before you start: Frequently Asked Questions

If you haven't purchased Unify yet, you may be here looking for specific information to help you make a decision. Head over to our Frequently Asked Questions page.

Problems?

If you have Unify already, but it's not working for you, go to our Problems and Solutions page.

Getting Started

- Downloading and Installing Unify
  - Uninstalling Unify, should you need to
- Getting started with Unify
- Navigating Unify's graphical user interface
- Loading and playing patches
- Working with plug-ins
  - Built-in and bundled plug-ins
  - Scanning and using your own plug-ins
  - Defining subsets for quick access to your favorite plug-ins
- Problems and Solutions

Creating new sounds

- Tour of the layer-stack view
- Tutorial: Creating your first patch
- Basic patch recipes

Controlling sounds through time

- Using Macro Kno s
- Using MIDI
- DAW automation

Technical details

- Understanding Unify's CPU meters
- How Unify processes MIDI data
- Show MIDI view
- Using the Transport
- Using the Settings view
- Adding new sound li raries

Using the bundled plug-ins suite

Instruments

- Sine Wave Synth (built-in) plays simple sine tones for testing
- Guru Sampler (built-in) our flagship sampler instrument
- 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
- PG-8X Roland JX-8p emulator

MIDI Effects

- MIDI Filter (built-in) multi-function MIDI event filter
- Ripchord Player (built-in) turns single key-presses into chords
- PolyBox (built-in) creates a virtual polysynth from multiple monosynth layers
- BlueARP sophisticated MIDI arpeggiator

Delay

- Omega Delay (built-in) creative stereo delay with integrated distortion and chorus
- mda Delay basic stereo delay
- mda DubDelay creative delay with feedback saturation and time/pitch modulation

Distortion

- NoizBox (built-in) creative distortion effect

- mda Bandisto multi-band distortion
- mda Degrade sample-rate reducer
- mda Overdrive warm tube-type distortion

Dynamics control

- 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
- 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

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

- KlangFalter open-source convolution reverb

Getting started with Unify: Overview

This page introduces some basic ideas about what Unify is and how it works, as well as some terminology concerning its graphical user interface (GUI). You can safely skip this material on first reading, and come back for a closer look as you get to know the program. To get started right away, go straight to Downloading and Installing Unify.

What is Unify?

Unify is both a stand-alone app and a plug-in (AU, VST, or VST3) that can load any Audio Unit (Mac OS only), VST or VST3 plug-ins to make new sounds. Unify can load MIDI Effects, Instruments (with keyboard note and velocity ranges) and Audio Effects into a “flat interface” where the most important functions are visible without diving into multiple pages or menus. 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 where you can apply Master effects.

Think of Unify as a super-powerful Combination mode like what is found in top of the line music workstations on the market… However, with Unify you can use ANY instruments and effects, not just the ones provided by a particular company.

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

Unify comes to you as one *stand-alone app* and two or three *plug-ins* (VST, VST3, and Mac Audio Unit) you can use in a DAW app like *Logic Pro*, *Cubase*, *Ableton*, *FL Studio*, etc. The app and plug-in versions are basically identical; the app version just gives you the convenience of playing without a DAW, e.g., for live performance.

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 VST/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 VST/VST3's and use them in *Logic*. **NOTE:** There are still a number of plug-ins that Unify cannot yet load. See the known issues page for details.

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.
- **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.

See full list of bundled plug-ins.

Unify is a platform for PlugInGuru sound libraries

Unify is also the future vehicle for most new *PlugInGuru.com* patch libraries. We plan to add new features and abilities to Unify and will release patch libraries that take advantage of the new abilities that a particular update will include.

Unify includes the Unify Standard Library–over 100 ready-to-play sounds using only bundled samples and plug-ins–and will be the basis for a new series of sound libraries from PlugInGuru, starting in 2020.

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-in*, to perform various manipulations on MIDI data
2. **Instrument layers** host at least one *Instrument plug-in* and a chain of zero or more *Audio-effect plug-ins*
    - Instrument plug-ins (e.g., software synthesizers) receive MIDI and generate stereo audio
    - Audio-effect plug-ins (e.g., filter, chorus, delay) process audio as it flows through them
3. **Audio Effects layers** contain only a chain of zero or more *Audio-effect plug-ins*
    - There is always 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.

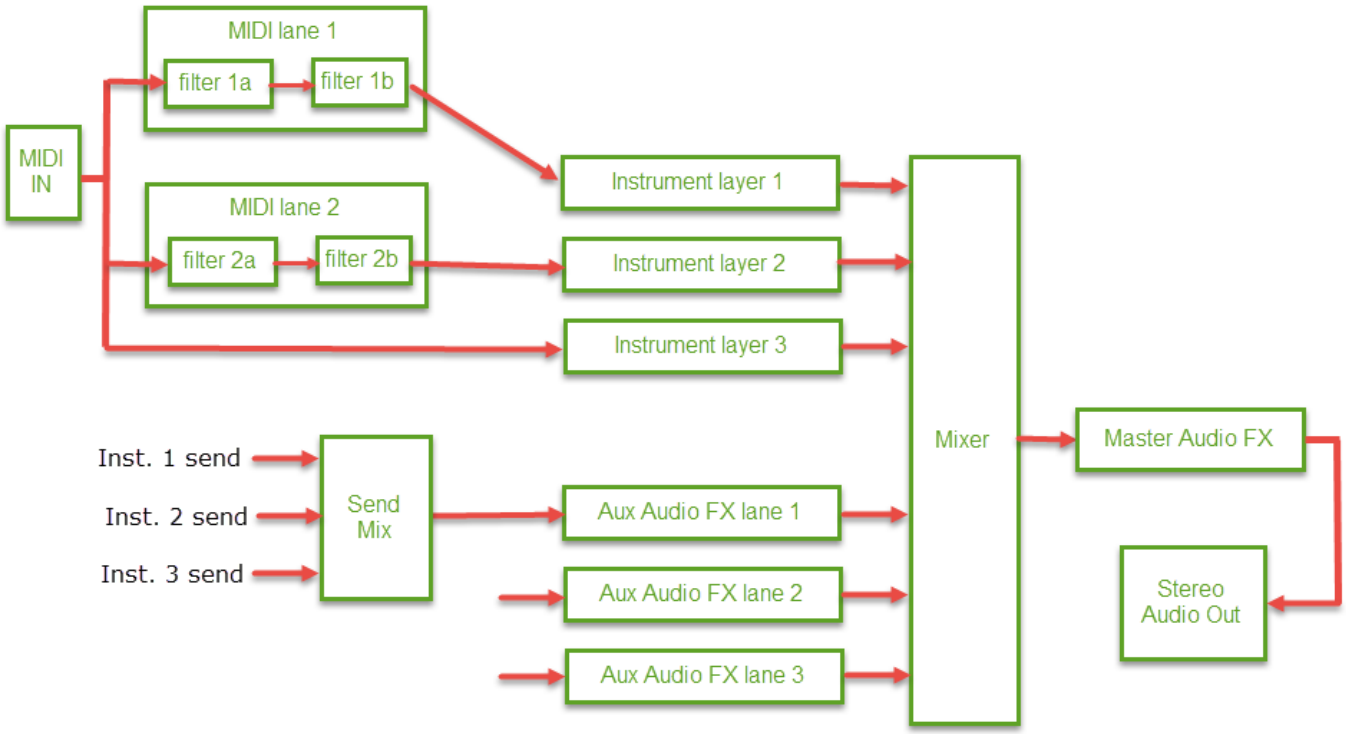
AUX Effects layers: Putting audio-insert effects on each individual Instrument layer can be useful, but often you would prefer to have just one or two effect chains which are shared among several Instrument layers. Unify provides *AUX Effects layers* for this. As soon as you create the first AUX Effects layer, Direct and AUX send controls appear on all Instrument layers, so you can separately control how much of that layer's output goes to 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 layers can perform certain manipulations of the input MIDI stream, even if no MIDI 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.

Frequently Asked Questions

If you're not ready to buy Unify, because you need some answers first, you've come to the right place. If your question isn't already listed below, contact us at UnifySupport@PlugInGuru.com and we'll add it!

What do I need to run Unify?

- **What kind of computer do I need?**
  - **Any recent Macintosh or Windows-based PC** should be fine.
    - **We are now compatible with MacOS 10.15 Catalina**
  - If your computer is powerful enough to run a DAW, it should run Unify.
  - You will need at least 2.5 GB of disk space for the Unify Standard Library.
    - The rest of Unify is just a couple of hundred megabytes.
  - Tons of RAM and a fast CPU with lots of cores are nice, but not required.
- **Do I need to own commercial plug-ins like Serum or Omnisphere?**
  - NO! You can get started making music with Unify without ANY other plug-ins!
  - IF you do own other commercial plug-ins like these, you can use them to create your own combination patches in Unify.

What do I get with Unify?

- **Unify stand-alone apps** for Mac and Windows
- **Unify plug-ins** for your DAW:
  - Windows: VST and VST3 formats (VST® is a trade mark of Steinberg Media Technologies GmbH)
  - Macintosh: Audio Unit, VST, and VST3
- **Built-in instruments and effects**:
  - *Guru Sampler* sample playback instrument
  - 16 custom effects: Delay, Distortion, Dynamics, EQ, Chorus/Phaser/Flanger, and four algorithmic Reverbs
- **Bundled third-party instruments and effects**:
  - Dexed emulation of Yamaha's classic DX7 FM synthesizer
  - Digits phase-distortion synth similar to Casio's CZ-series
  - OB-Xd subtractive analog-style synth, emulates Oberheim OB-Xa
  - PG-8X Roland JX-8p emulator
  - BlueARP MIDI arpeggiator
  - LoudMAX “brick-wall” limiter
  - KlangFalter convolution reverb
  - The complete set of classic/legacy MDA plug-ins
- **Unify Standard Library**
  - **Over 2 GB of samples** ready to play in *Guru Sampler*
    - Many adapted from previously published PlugInGuru libraries

        - Plus others from top sound designers (Airwave, Nato Feelz, Steve Taglione, and more)
    - **400 ready-made, ready-to-play combination sound patches**
        - By John “Skippy” Lehmkuhl and other top sound designers
        - All built using ONLY built-in/bundled plug-ins—Nothing else to buy!

Will ALL of my plug-ins work in Unify?

- **PROBABLY NOT**. Our beta testers and early-adopters have found several plug-ins that don't
- load properly in Unify, and some that will cause Unify (and your DAW) to crash.
  See Known issues and limitations in Unify 1.0
  If you have a favorite plug-in, and you know you'll be disappointed if it doesn't work, check with us first, before buying Unify.
  However, remember Unify comes with many built-in and bundled instruments and effects, over
- 2 GB of samples, and 400 awesome patches, ready to play, out of the box, so you might want to buy now anyway.
- We are going to be working VERY hard on plug-in compatibility issues from now on, so plug-ins
- that don't work today will work in Unify eventually.

Can I use Unify to load plug-ins my DAW won't support?

- **YES**. Unify works as a *bridge*, allowing you to load plug-in types which your DAW itself may not support.
  You **can** use Unify to load VST/VST3 plug-ins in Logic Pro X and GarageBand.
    - Provided your VST/VST3's are built for Mac
    - You **can't** load plug-ins built for Windows into a Mac host, or vice versa.
    - (*Unify PRO*, coming later in 2020, *will allow you to do this,* across a network.)
- You **can** load 32-bit VST plug-ins in Unify on a Windows PC using *jBridge*
    - Unify itself is 64-bit, but the Windows version includes built-in support for jBridge, an inexpensive 32/64-bit VST bridge program.
- You **can** load 32-bit Audio Unit and VST plug-ins on a Mac using 32 Lives
    - One of our beta-testers has verified this.
    - This will NOT work with *MacOS Catalina*, which has removed support for 32-bit binaries
-

Will Unify work in my DAW?

-

The plug-in versions of Unify should work in any DAW that supports VST/VST3 (Mac and Windows) or Audio Unit (Mac only) plug-ins, but in a few cases, a bit of setup may be needed.

Can I use Unify in Avid Pro Tools?

-

    - **NO**, not at this time.
    - Pro Tools uses its own proprietary AAX plug-in format, and cannot load VST/VST3 or Audio Unit plug-ins.
    - We hope to have an AAX version of Unify available at some point, but we can't predict when.
- Can I use Unify in Image-line FL Studio?
    - **YES**. Just make sure to **turn on "use fixed-size buffers"** for Unify.
- Can I use Unify in Steinberg Cubase?
    - **YES**.
- Have you tested every DAW under the sun?
    - **NO**, we can't afford to buy them all 😉
    - However, we are not YET aware of any DAW or plug-in host program that cannot run Unify.

Why isn't Unify perfect in every way?

- **PlugInGuru.com is a small company**. Just two people:
    - Veteran sound designer John "Skippy" Lehmkuhl and programmer Shane Dunne.
- **Plug-in technology is incredibly complex**, and lots of important details simply aren't published, so have to be figured out by trial and error.
    - Even Shane's PhD in computer science and 35+ years as a programmer, and John's 35+ years experience in sound design with Korg are no match for all the stuff that simply isn't written down anywhere!
    - Huge companies struggle with this stuff too; they just hide it better 😀
- **Life is short—make music!**
    - We didn't want to delay releasing Unify any longer, waiting for it to be "perfect".
    - People have been bugging us for months to be able to buy Unify and get started with it.
    - Software is never truly "finished", so why wait? Let's all get on with *making music!*

How can I share patches I create in Unify?

- Yes, but there are some things you have to do, to ensure it works.
- See Sharing Unify patches

Can I make patch libraries for Unify?

- Yes, but there are MANY things you have to do, to ensure it works.
- See Creating patch libraries for Unify

Downloading and Installing Unify

Getting started with Unify involves five major steps:

1. Purchase
2. Download the installer and the Unify Standard Library
3. Install Unify
4. Activate and Set Up Unify
5. Install the Unify Standard Library

Each of these major steps breaks down into smaller steps, as described in the sections below.

Purchase Unify

If you don't own Unify yet, you can go to PlugInGuru.com or to a number of online retailers that are also distributing Unify (NOTE: if you need technical support you will contact them for assistance). You'll find both a free DEMO and a for-purchase version.

You'll need to set up an account on the site even for the DEMO version - after registering and paying, you will receive an email which contains your license number and the links for the files required to operate Unify (NOTE: The links in the email last only 24 hours or 1x use - after that, please log into the site where you purchased Unify to download the links or see your license number).

Download the Installer and Unify Factory Content.guru file

You should receive *two download links* when you purchase Unify—one for the installer, and one for the Unify Factory Content.guru file. Go ahead and download both. That same email also has your License file to unlock Unify - this information is also available at the website where you purchased Unify (the files are online there as well).

The installer comes as a single file (*.dmg* file on Mac, *.zip* file on PC). The library file *UnifyFactoryContent.guru* is much larger (about 2.5 Gigabytes).

As soon as the installer download is complete, double-click the file icon to open it.

Before you Install Unify

Please be aware that the Unify Standard Library is approx 2.5 Gigabytes for the samples that come with Unify — so you should decide where you intend to keep it. The recommended location is inside the *PlugInGuru/Unify* folder inside your *Documents* folder. If you don't have enough room on the drive that contains your *Documents* folder, you'll need to put it on another drive. Also be aware future libraries will be adding more data to this folder so it will GROW over time if you add new libraries.

After deciding where to place the 2.5GB Libraries folder, you can begin the installation process!

On a Windows PC

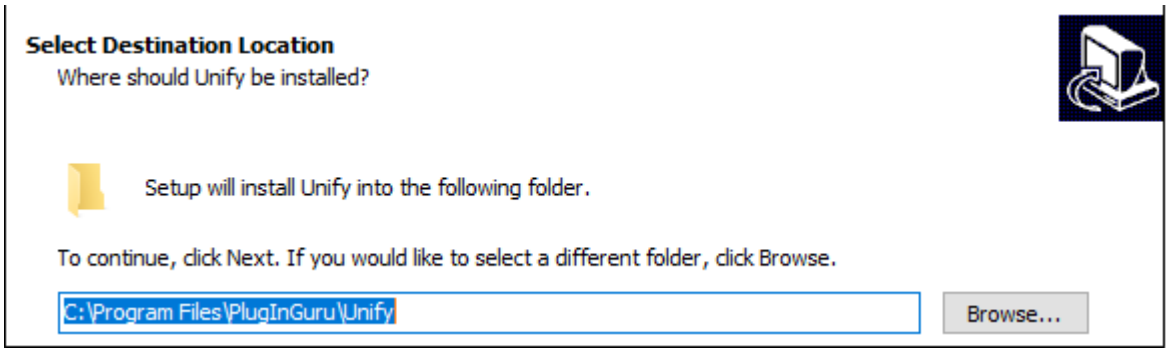
On a Windows PC, the installer file is called *InstallUnify_Win64_X.Y.Z.zip* (where *X.Y.Z* is the version number), and it contains three files:

- *UnifySetup.exe*
- *READ ME.rtf*
- A link to this Online Owners Manual

(You might not see the filename extensions *.zip*, *.exe*, and *.rtf*. Whether or not these appear is controlled by a setting in Windows itself.)

Please open the *READ ME* file before starting the installation, and at least skim over its contents. It may contain up-to-the-minute details which may affect how you do the installation.

VERY IMPORTANT: If you are installing Unify for the very first time, the installer will ask you to choose where you'd like to install Unify, as shown below. **PLEASE DO NOT CHANGE THE DEFAULT LOCATION**; doing so will cause all sorts of problems, because Unify won't know where to find bundled VSTs and its *PlugScanner* helper app.



Install Unify by double-clicking on the *UnifySetup* icon. You'll need to confirm that you want to run the installer, and then you'll see a pretty standard Windows installer wizard, which will require you to click “OK” a few times to accept the License Agreement and complete the installation. You will also be asked to specify where you prefer to keep VST plug-ins on your system.

You can also place a shortcut on your desktop for immediate access to the stand-alone version of Unify.

After the installer is finished, locate the *Unify.exe* stand-alone executable program or double-click the short cut link; you'll find the Unify.exe inside *C:\Program Files\PlugInGuru\Unify*. Double-click its icon to open the program, and carry on to the Activation and Set-up step below. (Alternatively, on recent Windows versions you can click on the Windows icon or press the *Windows key* on your keyboard, type Unify, and hit Enter to run the program.)

On a Macintosh

On a Macintosh, the installer file is called *InstallUnify_Mac_X.Y.Z.dmg* (where *X.Y.Z* is the version number); this is a Mac disk-image file. Double-click the file icon to open it. You'll see three files:

- *2 Install Files - One is a 1st Time Install and the other is an Updater if for those who already have Unify Installed and want to update to the latest version.* * READ ME.rtf *(Installation and*

other details) * a link to this online Owners Manual

Please open the *READ ME* file before starting the installation, and at least skim over its contents. It may contain up-to-the-minute details which may affect how you do the installation.

Install Unify by double-clicking on the *InstallUnify* icon - only install onto your system HD because important helper applications are installed. If you want to place the 2.5GB Library location to a different location, don't worry - that is decided LATER when you first run Unify for the first time.

After the installer is finished, locate the *Unify.app* stand-alone executable program; you'll find it in your Mac's main *Applications* folder. Double-click its icon to open the program, and carry on to the Activation and Set-up step below. (Alternatively, you can use *Spotlight*: press Command+Space bar, type Unify, and hit Enter to run the program.)

Complete the setup in the Unify stand-alone app

If you are installing a newer version of Unify, after having previously installed an older version, the stand-alone Unify app should run normally.

First-time installation step 1: Select Unify content folder

If you're installing Unify for the first time, the first thing you'll see is a message window like this, asking where you'd like Unify's main content folder to live. (Note this image was captured on a Windows PC; the default path on a Mac would be */Users/Shared/PlugInGuru/Unify*.)

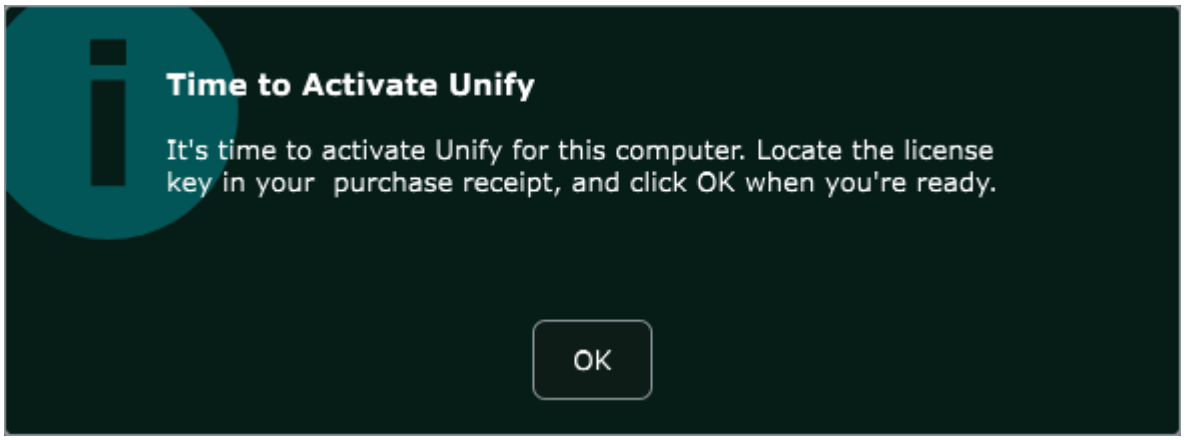
This folder can go wherever you like (and you can even move it later), but we recommend putting it on a disk that has at least 20 GB free space. You'll need less than 3 GB at first, but you'll probably want to add more Unify libraries later, so give yourself some room for growth.

If you click “Use default location”, Unify will automatically create the default folder. If you click “Let me choose”, a standard “choose-folder” dialog will appear, and you'll have to create the folder(s) yourself, then click “OK” to select the final folder, whose name might be, e.g., “PlugInGuru Unify Content”.

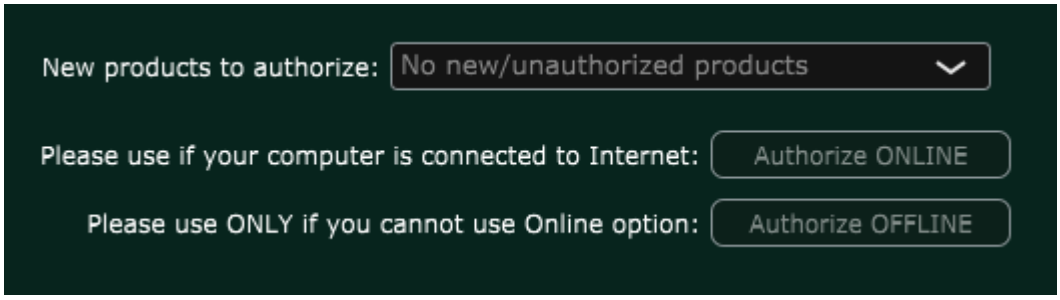
First-time installation step 2: Activate your license

As soon as you select the Unify content folder, you'll see another message window like this, advising

you to locate the License Code from your Unify purchase receipt, as you'll need it now, to activate Unify for use.



Click OK to dismiss the dialog, and you'll immediately see the full Unify window open to the Licensing view, with “Unify (Standard Edition)” already selected in the



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)

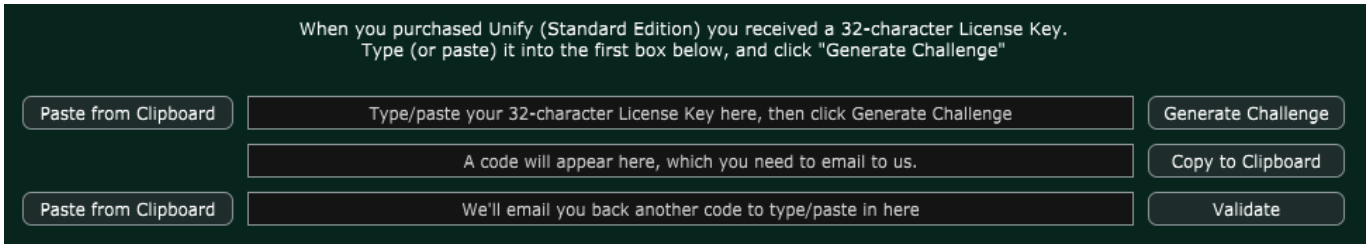
When you click the “Authorize ONLINE” button, the following should appear:

- 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 all goes well, 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 not, whatever message does appear should guide you in what to do next. Contact UnifySupport@PlugInGuru.com immediately if you need help.

Offline authorization

When you click the “Authorize OFFLINE” button, the following should 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 for you (e.g. copy/paste into a document on a USB drive), you need to get this code into a new email message and send it to 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, re-start 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 all goes well, you'll immediately be rewarded with a message in green text, saying “Valid license code accepted for Unify (Standard Edition). Thank you.”
- If not, whatever message does appear should guide you in what to do next. Contact UnifySupport@PlugInGuru.com immediately if you need help.

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* file, and get ready to drag its icon into the Unify window:

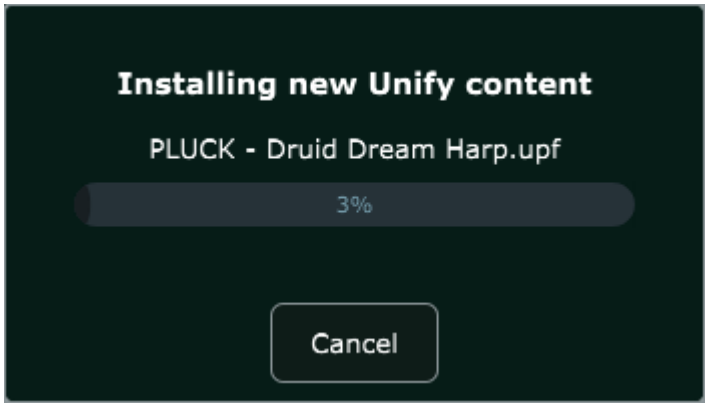
- If you **already downloaded** the *Unify Factory Content.guru* file, click the “I have it” button. A

standard file-open dialog will appear; locate the *Unify Factory Content.guru* file, select it, and click “Open”.
- If you **forgot to download** the *Unify Factory Content.guru* file, click the “Download it now” button. Your web browser should immediately begin downloading the file, and a message box will pop up with further instructions. Basically, you need to wait until the file is fully downloaded, then click the “OK” button. A standard file-open dialog will appear; locate the *Unify Factory Content.guru* file, select it, and click “Open”.

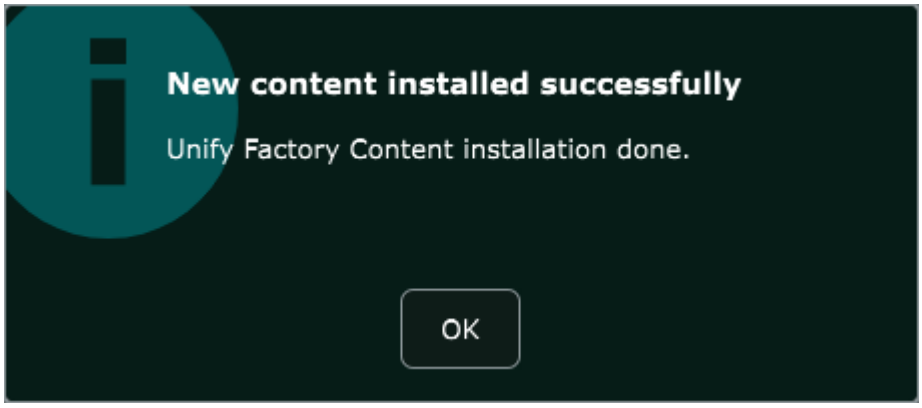
If you accidentally hit the “Cancel” button instead of “Open” when selecting the *Unify Factory Content.guru* file, just click the “gear” icon to go to Unify's Settings view, and **click the “Select .guru file…” button**. It's right in the middle—the only colored button on the page.

Whichever option you choose, once you select and open the *Unify Factory Content.guru* file, you will immediately see an information window like this:

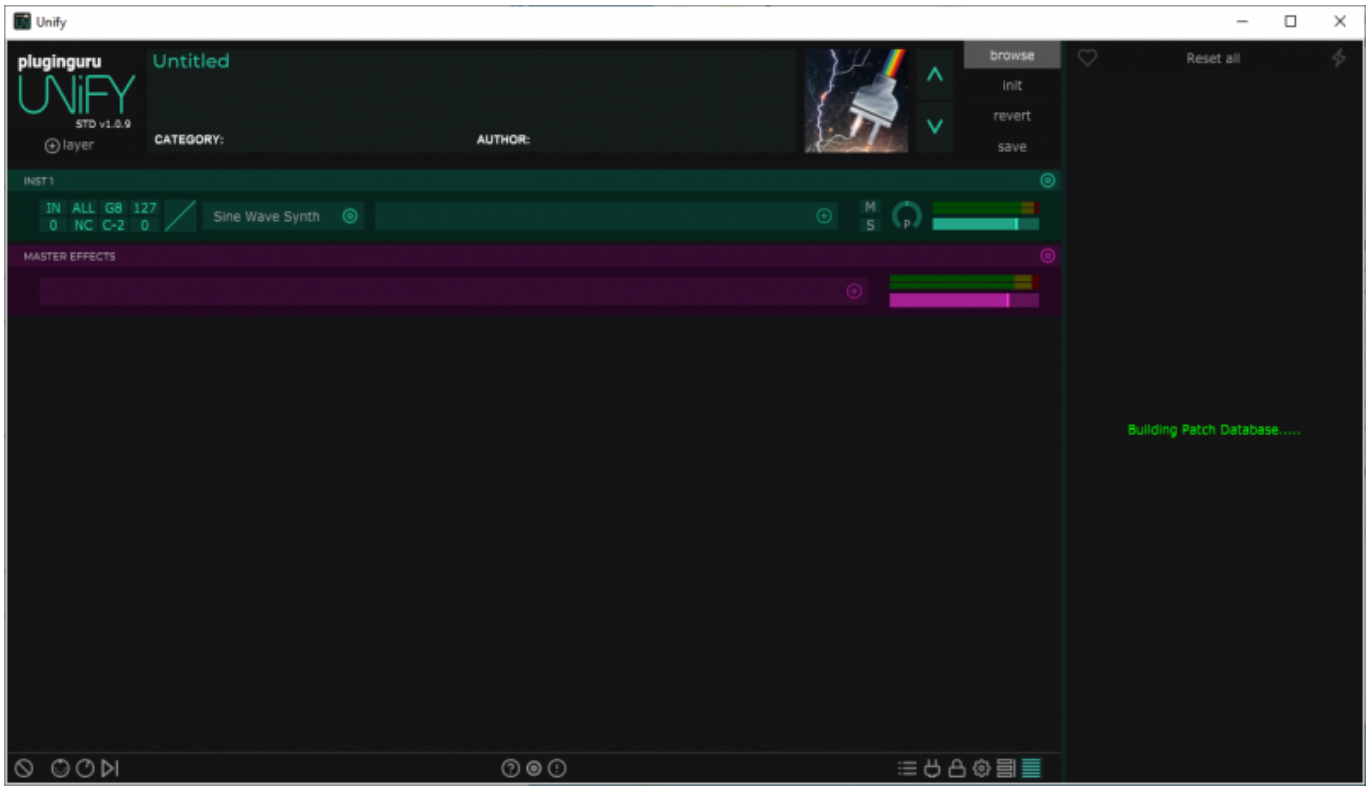
Click OK, and Unify will begin unpacking the factory content; you'll see a progress window like this:



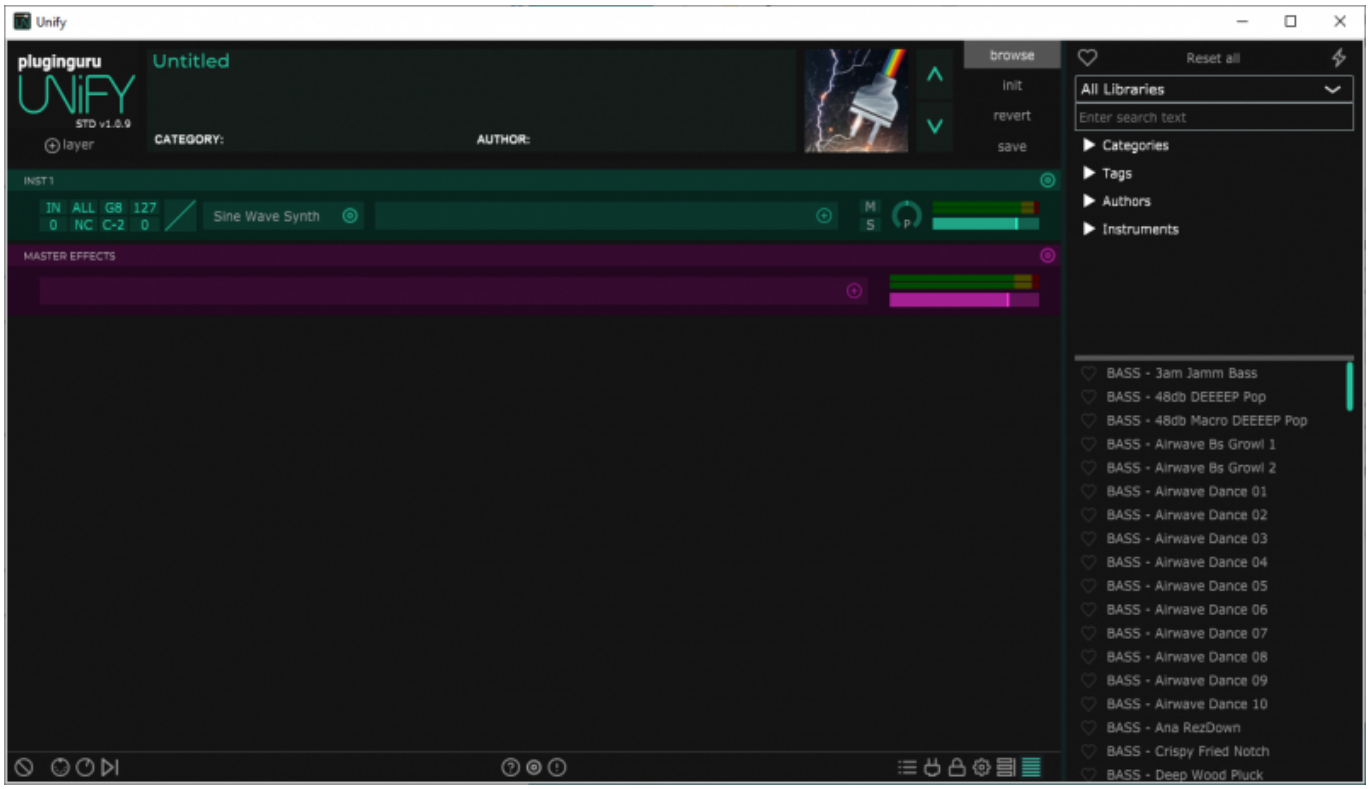
When it's done (this will take a while; there are a lot of files), you'll see the final confirmation window:



As soon as you click OK to dismiss this window, Unify's *patch browser* (sidebar) will immediately open, and you'll briefly see some green text saying that Unify is building its *patch database*.

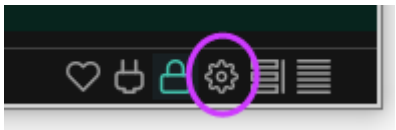


This may take anywhere up to about a minute, depending on the speed of your computer's disk drive, and when it's done, the list of patches will appear:

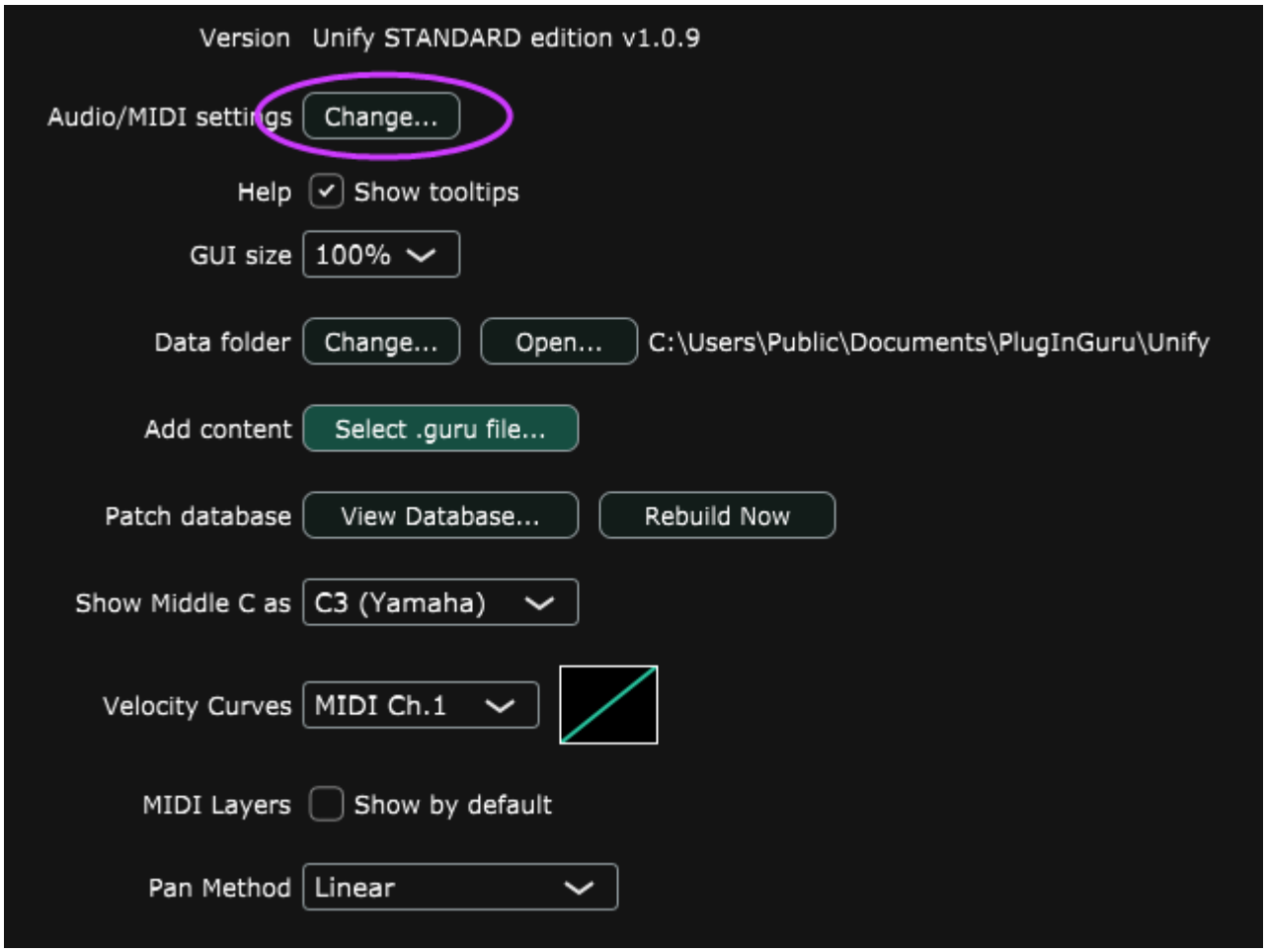


Final Set-up steps

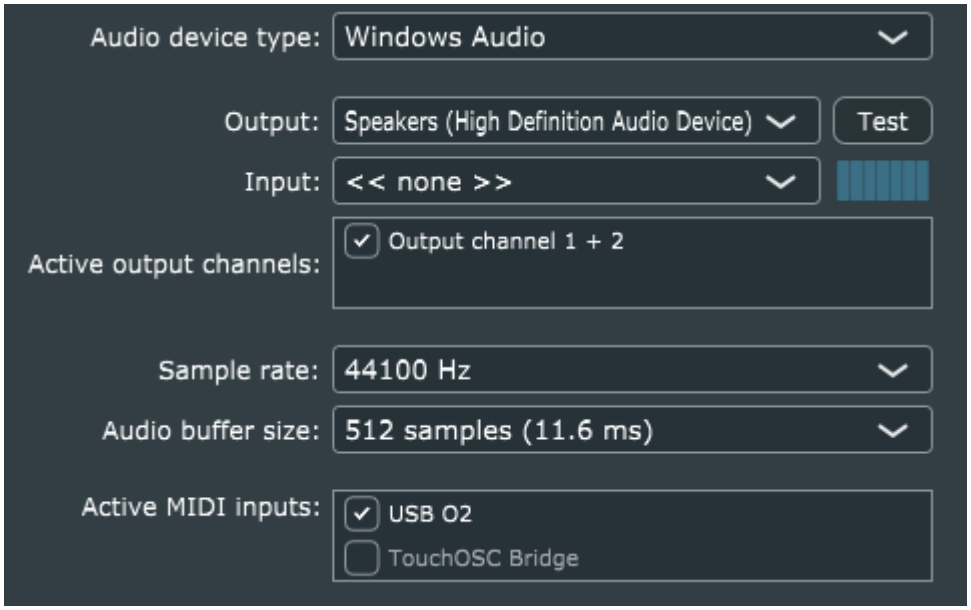
Before you can begin using Unify, you'll need to tell it about your audio and MIDI hardware. Look down to the bottom-right corner of the Unify window, locate the Settings icon (it looks like a gear), and click on it.



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



Click on the “Change...” button next to “Audio/MIDI settings” to open 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.)

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.

Before you get in touch, we'd like you to have a quick look for Unify's files and folders, to see if anything's missing.

- Click here if you have a Windows PC
- Click here if you have a Macintosh

Play!

Click on any patch in the list to open it and start playing with Unify! See Loading and playing patches in Unify for details.

Make patches!

If you'd like to get started making your own patches right away, go straight to Creating your first patch.

Uninstalling Unify

At some point, you may need to simply uninstall Unify completely from your system, so you can start over with a new install. The process is partly a manual one, and is slightly different for Mac and Windows machines.

See also Unify v1.0.6 files and folders for an overview of the files/folders you are about to delete.

Uninstall on Mac

There is no automatic uninstaller for the Mac version of Unify, so the entire process is manual.

1. Remove the main Unify data folder

- If you kept the Unify data (content) folder at its default location:
  - Navigate to */Users/Shared*
  - Drag the *PlugInGuru* folder to the Trash.
- If you moved the Unify data (content) folder somewhere else:
  - Check the folder location; it's displayed in the Unify Settings view.
  - Navigate to where the folder is in the Finder
    - In the Unify Settings view, click “Open…”
    - In the Finder window which appears, hold down Ctrl and click on the window title, then navigate UP one level to the folder that *contains* your Unify folder.
  - Close any running instances of Unify
  - Drag the folder to the Trash.

2. Remove the Unify app and plug-ins Note these are NOT in your own user folders.

- App: *Unify.app file* (may show as just *Unify* in the main */Applications* folder
- Audio Unit plug-in: */Library/Audio/Plug-Ins/Components/Unify.component*
- VST plug-in: */Library/Audio/Plug-Ins/VST/Unify.vst*
- VST3 plug-in: */Library/Audio/Plug-Ins/VST3/Unify.vst3*

You may be prompted to enter your password to confirm some or all of these deletions.

3. Remove the entire Unify settings folder

- Locate the *~/Library/Application Support/PlugInGuru* folder, and drag it to the Trash.

4. OPTIONAL: tell MacOS to forget earlier installs This step is almost never necessary. Please do this ONLY if directed to do so by PlugInGuru technical support.

- Open a Terminal window
- Type `sudo pkgutil –forget com.pluginguru.Unify` and hit Enter/Return
  - You will be prompted for your password
- Type `sudo pkgutil –forget com.pluginguru.UnifyData` and hit Enter/Return
- Type `sudo pkgutil –forget com.pluginguru.UnifyUpdate` and hit Enter/Return

Uninstall on Windows

1. Remove the main Unify data folder

- If you kept the Unify data (content) folder at its default location:
  - Navigate to *C:\Users\Public\Documents*
  - Drag the *PlugInGuru* folder to the Trash.
- If you moved the Unify data (content) folder somewhere else:
  - Check the folder location; it's displayed in the Unify Settings view.
  - Navigate to where the folder is in the Finder
    - In the Unify Settings view, click “Open…”
    - In the Explorer window which appears, navigate UP one level to the folder that *contains* your Unify folder.
    - (If you moved the entire *PlugInGuru* folder, navigate UP one more level to ITS parent folder.)
  - Close any running instances of Unify
  - Drag the folder to the Trash.

2. Use the Windows uninstaller

- Click on the Windows START button and type “uninstall a program”; hit Enter to open the Uninstall a Program dialog.
- Scroll through the displayed list to find the entry for Unify, and click (once) to highlight it.
- Click on “Uninstall” at the top of the window, and follow all on-screen instructions.

The uninstaller does a pretty good job of removing the Unify app and many other files, but won't remove any of Unify's data, including the primary data folder and the folder where its settings files are kept. The following steps allow you to ensure that nothing is left behind.

3. Remove the Unify app and bundled plug-ins/helpers Note these are NOT in your own user folders.

- Navigate to *C:\Program Files* and locate the *PlugInGuru* folder.
- If the folder isn't there, you're done. If it is, drag it to the Trash.

4. Remove the Unify plug-ins

- VST plug-in: *C:\Program Files\Common Files\VST2\Unify.dll*
- VST3 plug-in: *C:\Program Files\Common Files\VST3\Unify.vst3*

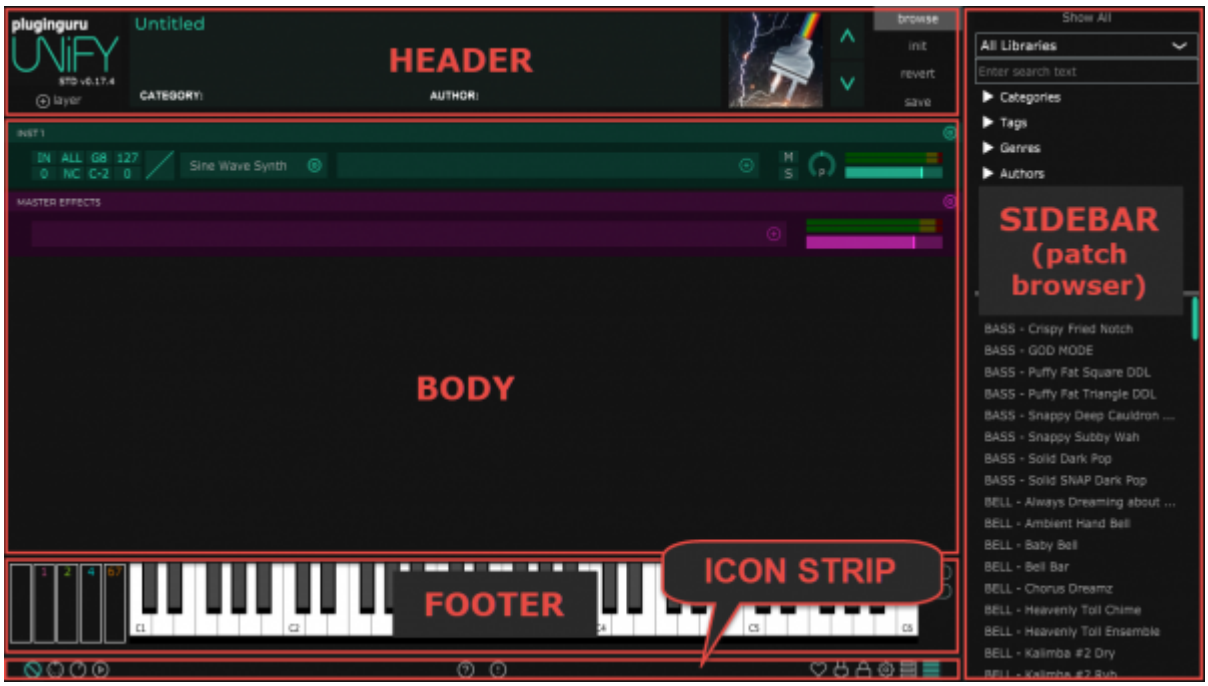
You may be prompted to enter your password to confirm some or all of these deletions.

5. Remove the entire Unify settings folder

- Locate the *C:\Users\<your username>\AppData\Roaming\PlugInGuru* folder, and drag it to the Trash.

Navigating Unify's graphical user interface

Unify's graphical user interface (GUI) window is divided into five main regions, as shown below:



- The **Header** displays information about the current patch, and contains several buttons for editing it. See Tour of the Header section below.
- The **Body** usually contains the *layer stack* for the current patch, but may also contain other views
    - Select views using the icons at the bottom right.
- The **Footer** usually doesn't appear at all, but it may contain certain auxiliary views, such as the *MIDI Monitor* view as shown in the picture above.
    - Use the icons at the bottom left to choose the Footer contents.
- The **Icon Strip** along the bottom edge of the GUI are described in the next section.
    - These mainly allow you to navigate by revealing different views.
- The **Sidebar** contains Unify's *patch browser*.
    - Click on the “browse” button at the top-right corner to open and close the sidebar.

The Body and Footer can each contain any of several views, which are described in detail on separate pages:

- Layer stack view
- Settings view
- Licensing view
- Plug-ins view
- Plug-in Subsets view

The Footer can also contain several views:

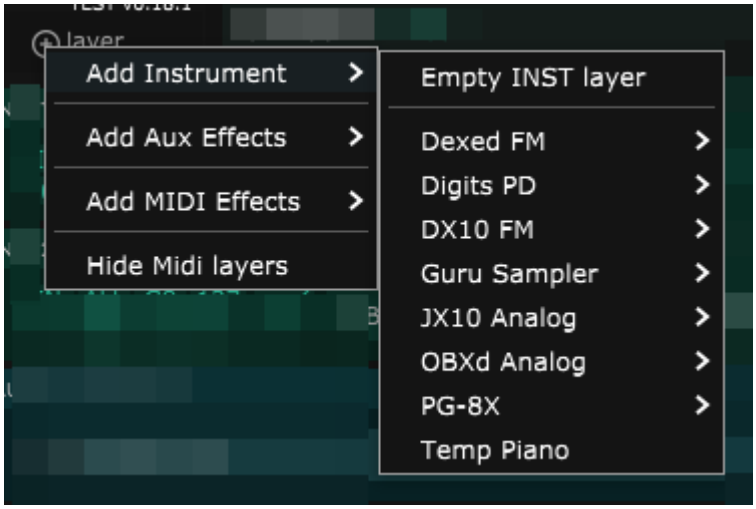
- The ShowMIDI view shows MIDI and CC activity
- The Macro knobs view is for *real-time MIDI control* and *host automation*
- The Transport view allows setting BPM tempo and starting/stopping Unify's *transport* (internal timing system)

Tour of the Header section

The Header section comprises many distinct displays and buttons:

The large **patch metadata area** describes the currently-loaded patch. If there are more than two lines of (white) descriptive text, you will see a small green **scroll-bar** at the right-hand end. You can scroll down to see the rest of the description either by dragging the scroll-bar with your mouse, or using your mouse's scroll wheel (or ball, or touch surface, or whatever scroll mechanism your mouse or trackpad offers) while the mouse pointer is anywhere inside the metadata area.

The **layer button** just under the Unify logo on the left opens a menu of *layer operations*. The following screenshot shows how this menu looks when the first sub-menu is open:



The first three items on the layer operations menu allow you to add new layers, either empty ones, or prebuilt ones chosen from various sub-menus of *layer presets*.

The **Show/Hide Midi layers** item selects whether MIDI layers (if present) are shown full-size, or shrunk down to just their titles to make more space for other layers.

To the right of the patch metadata area you'll see the **library image**, which indicates pictorially, which Unify library the current patch is part of.

To the right of the library image are the **prev/next buttons**, for switching to the *previous* or *next* patch in the current patch list (shown in the **patch browser** in the sidebar, if open).

Down the right-hand side of the header are four buttons:

- The **browse** button opens and closes the sidebar
- The **init** button clears (initializes) the current patch, for when you want to start making a new

one from scratch.
- The **revert** button re-loads the current patch from the last saved version on disk, for cases where you have made several changes, but don't want to save them.
- The **save** button is for saving the current patch to disk. See Creating your first patch.

Icons in the Icon Strip

The **Icon Strip** along the bottom edge of the GUI contains three groups of icons:

 The **icons on the left** allows you to select what appears in the Footer. From left to right:

- MIDI Monitor: shows MIDI activity coming into Unify from outside
- Macro Knobs: shows eight “macro parameter” knobs which you can link to multiple parameters
- Transport: shows Unify's *Transport*, which controls BPM timing
    - In the present version of Unify, this only appears in the stand-alone app

The active icon will turn green; click it again to close the Footer.

 The **icons on the right** allow you to select what appears in the Body. From left to right:

- *List icon:* (favorite plug-ins) opens Unify's *plug-in subsets view*
- *Plug icon:* opens the *plug-in scanning view*
- *Lock icon:* opens the *Licensing view*
- *Gear icon:* opens the *Settings view*
- The two rightmost icons both select Unify's *layer-stack view*
    - The one on the right selects “accordion view”, where all layers will be shown, but may be squashed vertically
    - The one on the left selects “scrolling view”, where all layers are shown at full height, and you may need to use the scroll-bar on the right (or your mouse's scroll wheel) to bring them into view.
- NOTE if you use any of the four leftmost icons, you can simply click the icon again to return to the *layer-stack view*

 The **three icons in the middle** are all different:

- The *question mark icon* brings up Unify's *Help menu*, which allows you to turn *tooltips* (helpful text hints which appear when you roll your mouse pointer over various parts of the GUI) on or off, display the *About box*, or open any of several Unify-related pages in your web browser.
- The *operations button* (two concentric circles) pops up a small menu of operations affecting Unify as a whole, e.g. changing the default window size.
- The *exclamation mark icon* is Unify's “MIDI Panic” button. Click this to immediately silence all plug-ins.

Loading and playing patches in Unify

Loading and playing patches in Unify *should* be as simple as clicking on the patch name in the patch browser. This page also covers how this might go wrong, and how to recover.

I don't see ANY patches!

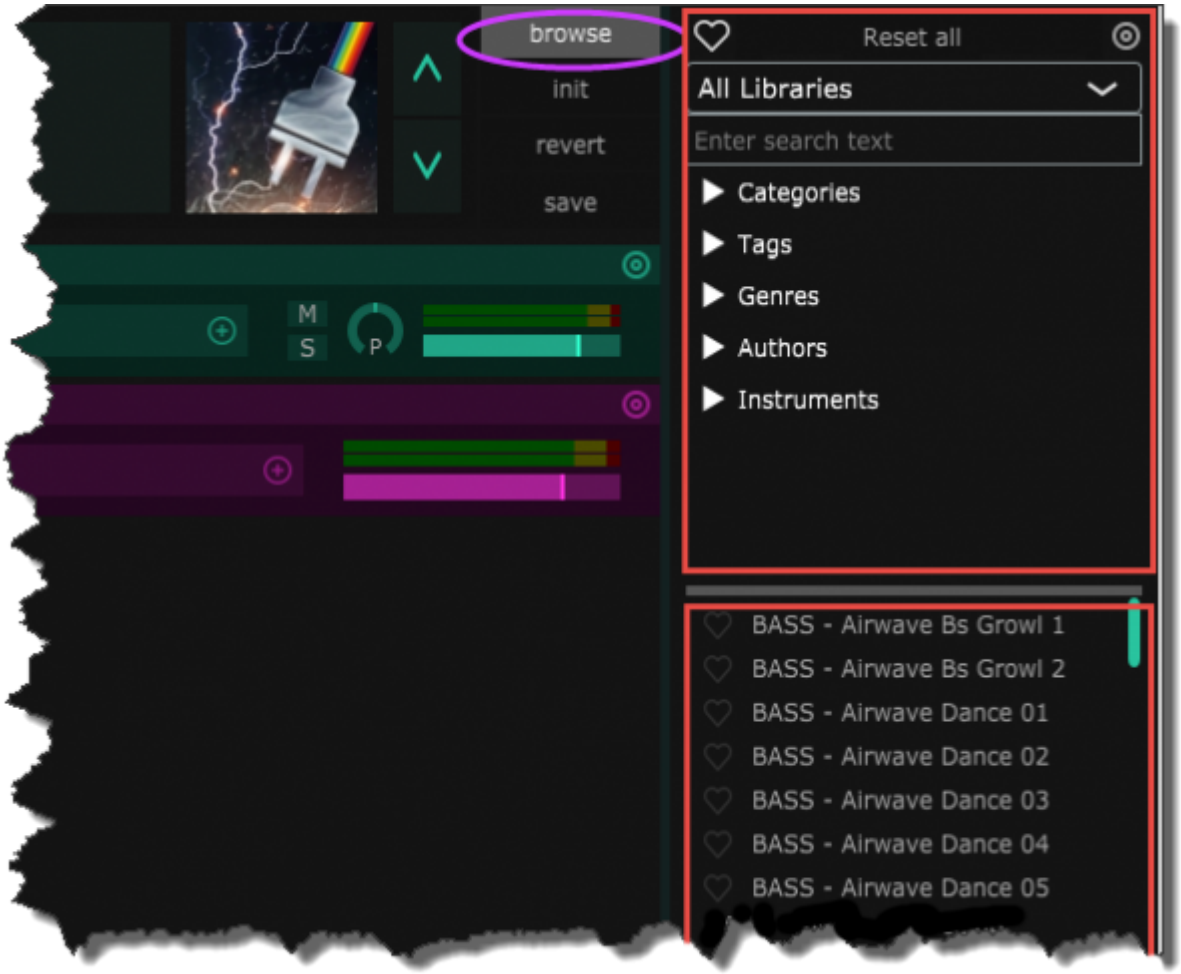
If you open the patch browser (see next section) and don't see any patches, one of three things may be wrong:

1. You haven't installed the Unify Standard Library yet: check these instructions
2. The path to the Unify data folder on the Settings page might be incorrect: check these instructions
3. You need to rebuild the patch database: click the “gear” icon at bottom right to open the Settings page, and click the “Rebuild Now…” button.

If you see some patches, but fewer than you expect, you almost certainly need to rebuild the patch database.

Using the Patch Browser (sidebar)

Unify's *patch browser* view lives in the sidebar extension on the right-hand side of the GUI, which may not always be visible. Click the **browse button** to open and close it:



The patch browser consists of two main sections:

1. The top section (outlined in red in the picture above) is called the *Selector*, and contains controls to allow you to narrow down the available choices in the bottom section.
2. The bottom section (also outlined in red) is called the *Chooser*; it contains a list of patch names in alphabetical order. Click once on any patch name to load that patch.

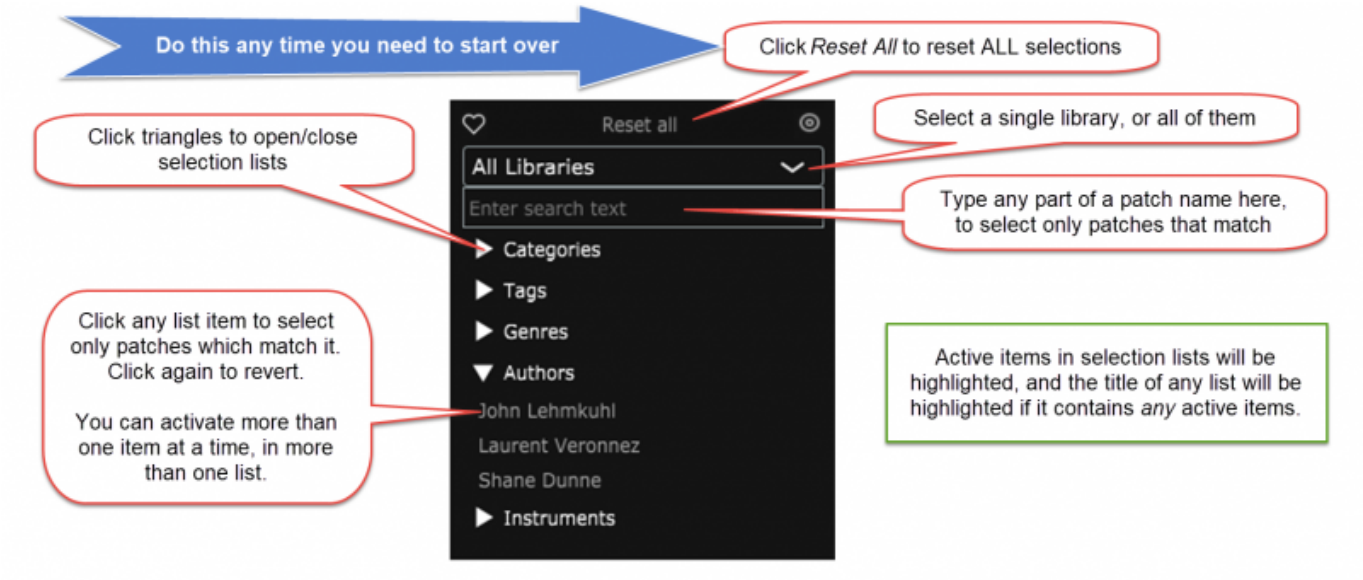
In addition to these two main sections are three less-obvious things you need to know about:

1. Between the Selector and Chooser is a grey “split bar” which you can drag to adjust the relative heights of the two main sections.
2. To the left of the sidebar itself is another split bar (like the first one, it lights up bright blue when you roll the mouse pointer over it), which you can drag to adjust the width of the sidebar.
3. To the left of the **browse button** are two buttons with up- and down-pointing green chevrons (see below)
    - Click the downward-pointing chevron button to load the *next* patch listed in the Chooser
    - Click the upward-pointing chevron button load the *previous* patch

Using the Patch Selector

When you first open the patch browser, the Chooser (bottom part) will show *all patches on your system*, which may run to many hundreds once you have more than a few libraries installed.

The controls in the Selector (top part of the sidebar) allow you to narrow down the set of patches in the Chooser, to help you find what you want quickly:



The icons at the top-left and top-right are as follows:

- Click the “heart” icon to toggle showing only patches you have marked as “favorites” (by clicking the heart icon next to the patch name in the Chooser section).
- The “ops button” icon (two concentric circles) pops up a small menu of database-related operations. As of Unify v1.0, this has only one item: rebuild patch database.

About the Patch Database

All the controls in the Patch Selector make use of Unify's *patch database*, which is basically an index to all the patches on your system.

- To see the how the database is structured, open the Settings view by clicking on the “gear” icon at the bottom right, and click the “View Database…” button.
- For every patch, the database keeps track of the patch name, the file name (which *need not be the same* as the patch name), what folder it is in, which library it's part of, and lots of other details.
- Whenever you add patches or libraries to your system, Unify needs to *rebuild* the patch database by re-scanning all the patch files. This happens automatically when you open Unify (you might notice a short delay), and you can force a rebuild at any time by clicking the “Rebuild Now…” button in Settings view.

Working with plug-ins

Unify doesn't produce any sound by itself. It relies on *plug-ins* to produce and modify sound. Unify includes a substantial set of plug-ins, some of which are actually built into the program, and can also work with just about any other plug-ins you own.

Types of plug-ins

Unify can deal with three **categories** of plug-ins:

- **Built-in** plug-ins are part of the Unify program itself
- **Bundled** plug-ins include free and/or open-source instruments and effects, whose authors/developers/maintainers have given us permission to distribute with Unify.
- **Other plug-ins you own** includes just about all the plug-ins now on your system, and others you may buy later.

Click here for the full list of plug-ins included with Unify.

As you may know, plug-ins also come in various, mutually-incompatible **formats**. Unify can load any of the following three formats (Audio Units is Mac-only):

- **VST** (“virtual studio technology” was developed by (and remains a registered trade mark of) Steinberg Media Technologies GmbH.
- **VST3** (VST version 3) is a newer format also developed by Steinberg.
- **Audio Units** (AU) is a format developed by Apple Corporation.

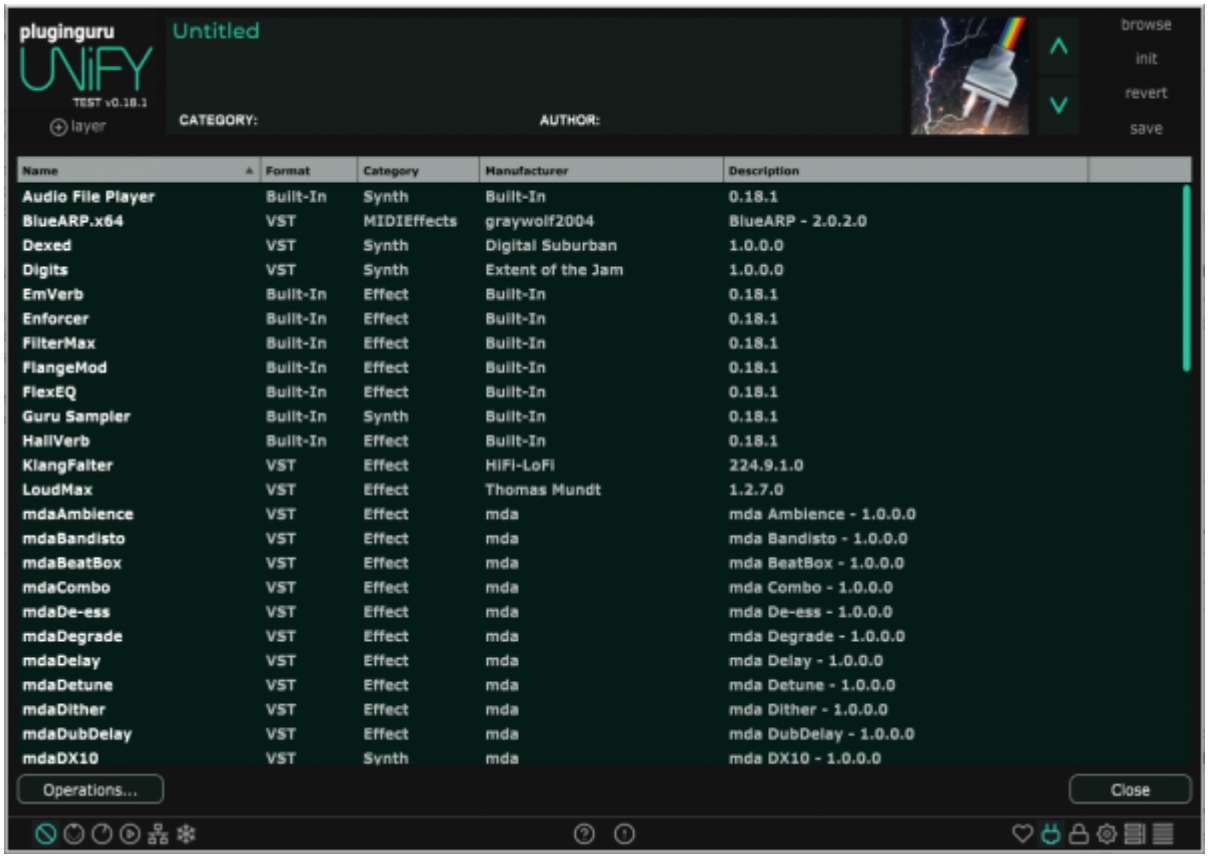
There are a handful of other formats which Unify cannot use, most notably AAK (Avid Audio eXtension), a proprietary format developed by Avid Technology, Inc. for use with their Pro Tools DAW.

Unify itself comes as a stand-alone app and as a plug-in, in VST, VST3, and Audio Unit formats. (An AAX version of Unify may be available at some time in the future, but even it would not be able to load AAX plug-ins.)

Unify's plug-in database

Unify maintains a simple database of plug-ins it knows about, which includes details like the plug-in name, manufacturer name, format (e.g. VST, AU, etc.), and the path to the folder where the plug-in is stored.

You can inspect the contents of this database by clicking on the *plug icon* near the bottom-right corner of the Unify GUI (highlighted in blue-green in the following screenshot). This is called the **plug-ins view**. (Note some details, such as folder paths, don't appear in this view, but they are still present in the database.)



This plug-in database is automatically populated with details about Unify's built-in and bundled plug-ins, but other plug-ins on your computer must be registered with Unify, so it can add them to its database. This can be done in either of two ways:

1. **Drag-and-drop:** To register one or a few plug-ins at a time, locate the plug-in file(s) and drag its icon directly onto Unify's plug-ins view.
2. **Scan:** You can also have Unify *scan your system* to locate plug-ins, as described in the next section.

Built-in and Bundled plug-ins

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.

MIDI Effects

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

- **MIDI Sustain** (built-in) makes the sustain/damper pedal (MIDI CC#64) work for plug-ins that don't recognize it.
- BlueARP, a wonderful VST arpeggiator by developer Oleg Mikheev aka *greywolf2004*
- Basic MIDI transpose, filter, and latch functions are built into every MIDI and Instrument layer

Instruments

Instrument plug-ins receive MIDI and produce audio.

- **Sine Wave Synth** (built-in) plays simple sine tones for testing
- **Guru Sampler** (built-in) is our flagship instrument, which lets you use whole libraries of samples developed by John Lehmkuhl and others
- Dexed is an open-source emulation of Yamaha's classic DX7 FM synthesizer
- Digits is an open-source phase-distortion synth similar to Casio's CZ-series instruments, from developer Louis Gorenfeld.
- **mda DX10** is a basic 2-operator FM synth with very low CPU usage
- **mda JX10** a 2-oscillator analog subtractive style synth, also very easy on CPU
- OB-Xd is an open-source subtractive analog-style synth which emulates the classic OB-Xa by Oberheim, kindly shared by project maintainer George Reales.
- PG-8X is an emulation of the classic Roland JX-8p subtractive analog synthesizer, from developer Martin Lüders.

Audio Effects

Audio effect plug-ins process audio in various ways, and can be chained together. Unify includes many such plug-ins, which fall into six categories:

Delay

- **Omega Delay** (built-in) is our own creative stereo delay with integrated distortion and chorus
- **mda Delay**, basic stereo delay

- **mda DubDelay**, creative delay with feedback saturation and time/pitch modulation

Distortion

- **NoizBox** (built-in) is our own creative distortion effect with user-definable response curves
- **mda Bandisto** multi-band distortion
- **mda Degrade** sample-rate reducer
- **mda Overdrive** warm tube-type distortion

Dynamics control

- **Enforcer** (built-in) is our own stereo compressor
- **Pump House** (built-in) adds tempo-locked pulsation to any sound with user-definable volume curves
- LoudMAX is a “brick-wall” limiter from developer Thomas Mundt
- **mda Dynamics** compressor/limiter/gate
- **mda Envelope** envelope follower / VCA
- **mda Limiter** optoelectronic-style limiter
- **mda MultiBand** multi-band compressor with mid-side processing modes
- **mda Splitter** frequency/level crossover

Equalization and Filtering

- **Restrictor** (built-in), our own band EQ with independent control over lower and upper bounds
- **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), a beautiful clean stereo chorus
- **PhaseMod** (built-in) stereo 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

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
- KlangFalter is a terrific open-source convolution reverb from developer Uli Haberhauer.

Scanning and using your own plug-ins

Before you can use any of your own plug-ins with Unify, they must be registered in Unify's plug-in database. You can register plug-ins individually, or you can have Unify scan your system for available plug-ins.

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”

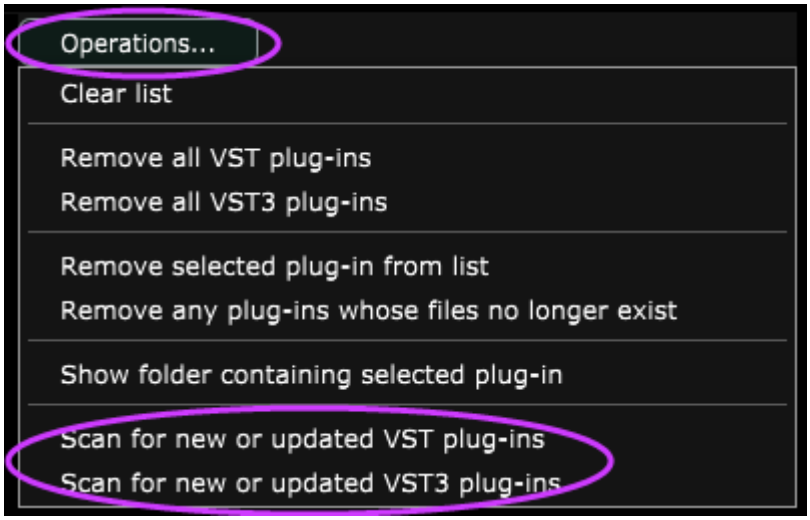
Scanning your system for plug-ins

You can also have Unify scan your system for available plug-ins. Note that when when scanning for plug-ins, Unify actually has to try to load each one. This will certainly trigger any copy-protection mechanisms it uses, and in some cases, may even cause Unify to crash. We therefore recommend the following “pro tips”.

Pro tips for scanning

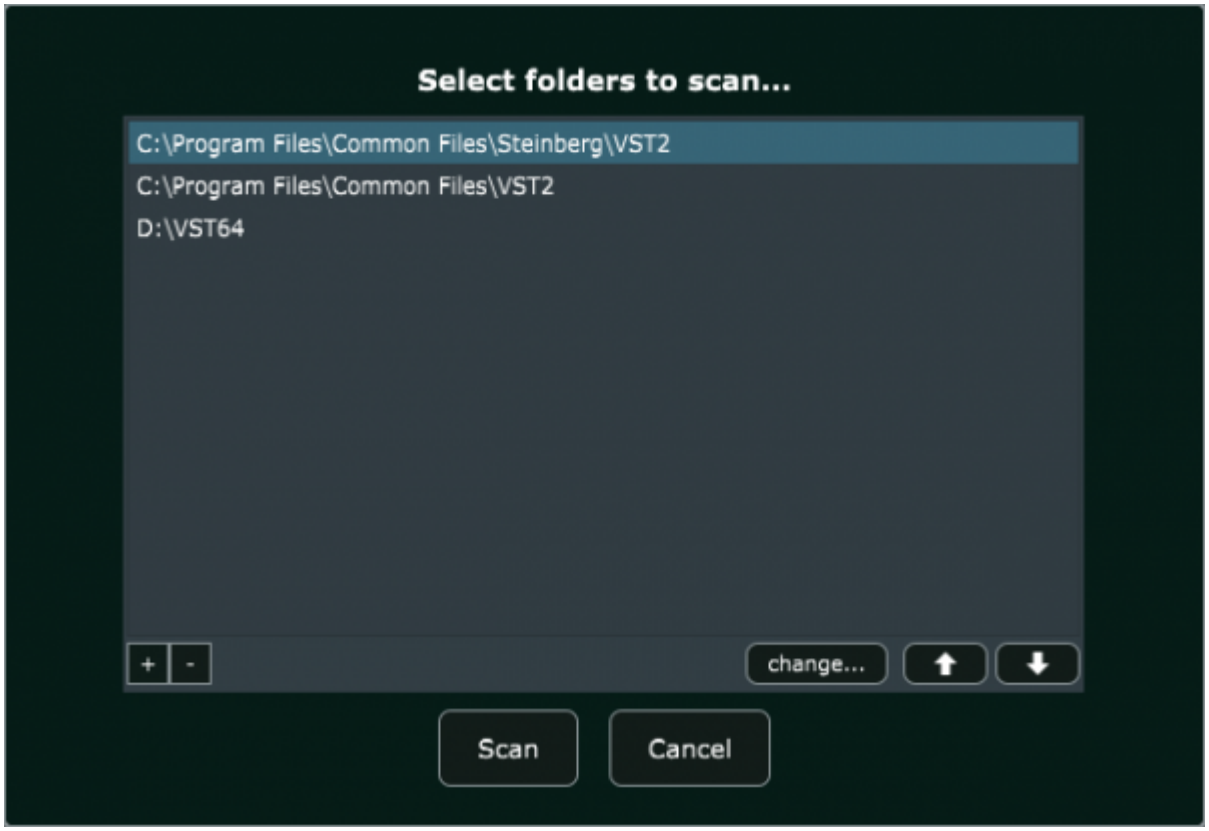
- **Scan using the stand-alone Unify app**, not in your DAW.
- **Avoid scanning your entire system** if you can. It's perfectly OK to scan just one or a few folders at a time, and you can always come back another day to scan some more.
- **Stay near your computer**, so you can be ready to deal with e.g.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



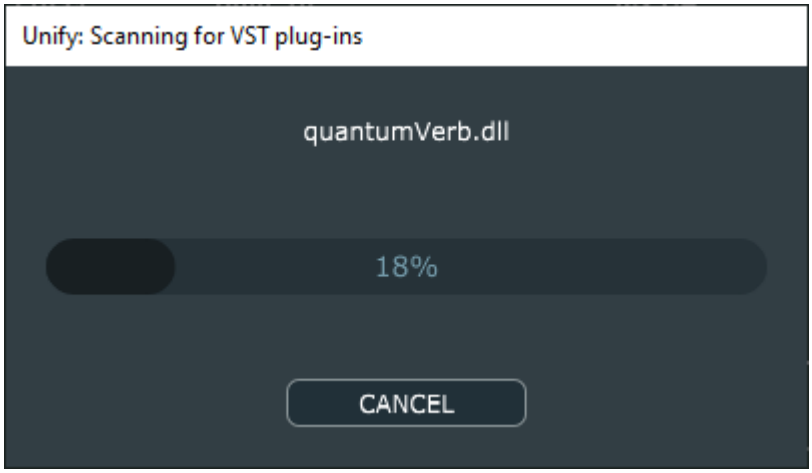
The VST and VST3 standards do not specify exactly where plug-ins ought to be located, and many systems (especially Windows systems) may use quite a few different plug-in folders. *It's up to you to know where these are, and how to specify the paths to them,* just as you do for working with plug-ins in 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 re-start it, and it should continue from where it left off. If you 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.

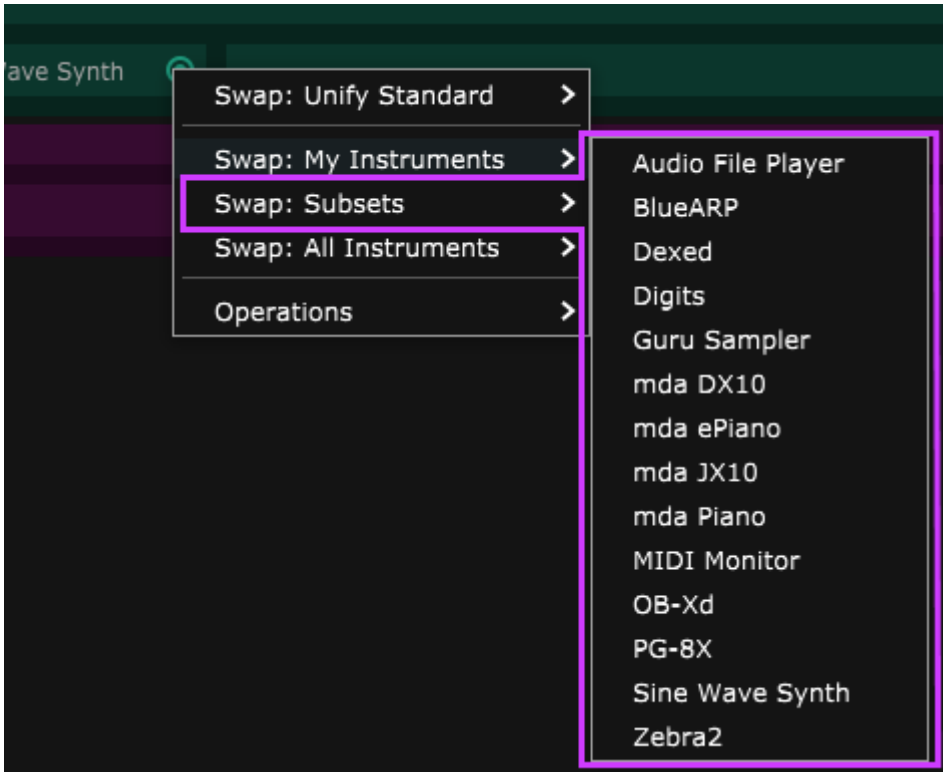
- On a Mac, you may see a pop-up saying that some plug-in needs to access information from your keychain, and you must enter your password to allow this. After entering your password, **we advise that you click “Always Allow” instead of “Allow”**, because you will probably have other plug-ins from the 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.

Defining subsets for quick access to your favorite plug-ins

When you have dozens of plug-ins, choosing them can be difficult. Unify gives you the ability to define your own **plug-in subsets** which automatically show up in selection menus like this one:



Pre-defined subsets let you get started right away

Unify comes with several pre-defined subsets (which you can edit if you wish):

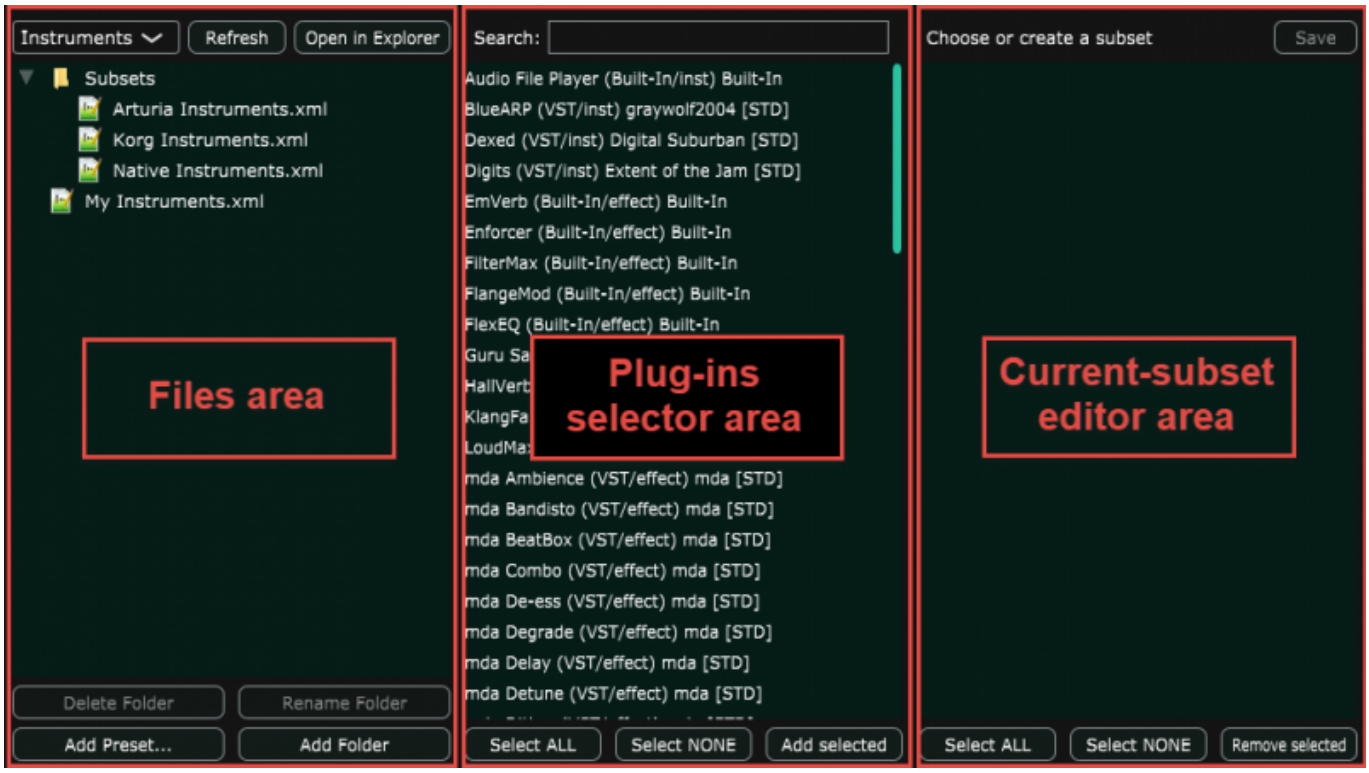
1. Every plug-in selection menu begins with a “Unify Standard” item, allowing you to select among Unify's built-in and bundled plug-ins.
2. For each category of plug-ins (MIDI effects, Instruments, Audio effects), there is also a pre-defined “My…” subset, which is stuffed with references to hundreds of common plug-ins, but *only plug-ins which are actually on your system* (and have been registered with Unify's plug-in database) will actually appear in pop-up menus.
3. Every plug-in selection menu also contains an “All…” sub-menu, which lists ALL available plug-ins (in the relevant category), using one sub-menu for each manufacturer name, just as you'll see in most DAWs.

Using the plug-in subsets view

Thanks to these pre-defined subsets, you can work with Unify without ever defining or editing your own subsets, but eventually you will probably want to. Get started by clicking the *heart icon* at the

bottom right part of Unify's icon strip:

The following *plug-in subsets view* will then appear in Unify's body area. It can be a little daunting at first glance, but begins to make sense when you see that it consists of three distinct *areas* or *columns*:



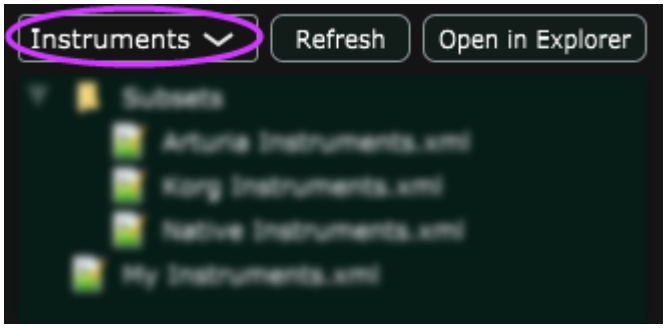
The basic concept involves four steps:

1. You choose (or create) a subset file in the *Files area* (on the left)
    - You can also use the controls in the Files area to rename or delete presets
2. Its contents appear in the *Current-subset editor area* (on the right)
    - You can then *delete any plug-ins you don't want* by selecting them and clicking “Remove selected”
3. You can then *add new plug-ins to the preset* by
    - locating and selecting plug-ins you want in the *Plug-ins selector area* (in the middle), and
    - clicking “Add selected” button to add them.
4. When you're done, you *save the edited preset* by clicking the “Save” button (top right corner)

Files area: Selecting, renaming and deleting subsets

Unify uses a hierarchical system of XML files in folders to remember your plug-in subsets. (If you'd like to look around these files, go to the Settings view by clicking on the “gear” icon, click the Data folder “Open…” button, then open the *Presets* folder, and finally the *Plugin Subsets* folder inside that.)

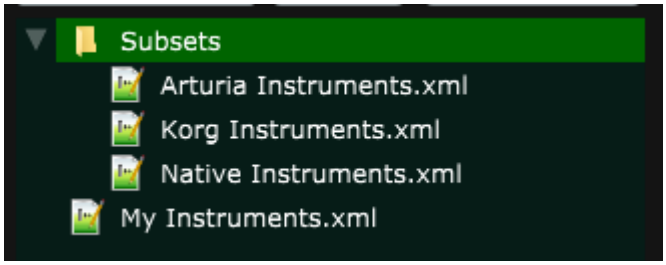
The top level of the subset hierarchy is the three basic **plug-in categories**: *Instruments*, *Audio effects*, and *MIDI effects*. The pop-up menu at the top-left corner of the Files area lets you choose which one category you want:



When you select a category, the main part of the Files area shows the contents of the corresponding folder, consisting of *XML files* (individual subsets) and, in some cases, *sub-folders*. If you wish, you can click the “Open in Explorer” button (“Open in Finder” on Mac) to open the folder itself. **If you make any changes in the Explorer/Finder**, click the “Refresh” button to update the Files area view.

Selecting and de-selecting files and sub-folders

In the Files area view itself, you can **click any item to select it**, and use the small triangles to the left of each sub-folder to **show or hide the sub-folder contents**. Selected items will be highlighted in green like this:



When no items are selected, the surrounding folder (*Instruments*, *Audio Effects*, or *MIDI Effects*) is effectively selected. You can **use the “Refresh” button to clear selections** to get back to this state, e.g., if you want to create a new subset in that folder.

Creating a new subset file (or sub-folder)

To create a new subset file (or a sub-folder), you must first **select the folder you want to create it in**.

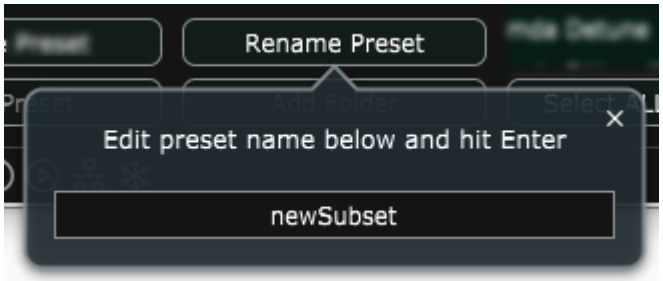
- Use the “Refresh” button at the top if you want to clear all selections and select the top-level folder.
- Otherwise, click the sub-folder in which you want to create the new item
- The new item will be created immediately, with a default name (“newSubset.xml” or “New Folder”). See below for how to rename it (or delete it, if you created it by mistake).

After creating and renaming a new preset file, you'll most likely want to select and add presets to it (below).

Renaming a preset file or sub-folder

At the bottom of the Files area are four buttons. The top two (“Delete” and “Rename”) will change, depending on whether you select a *file* or a *sub-folder* in the files-hierarchy display. (Remember, when *nothing is highlighted*, the surrounding top-level folder is effectively selected.)

The **Rename button** allows you to change the name of the selected file or folder. Clicking it pops up a small window like this:



- You **must click in the black edit area** to begin editing; then you can type
- You can use your mouse to position the edit cursor, or select characters to delete/copy/paste, etc.
- You **must press the Return/Enter key** on your keyboard for the change to take effect.
- To **cancel and close the window without renaming**, click the little “X” at the top-right corner, or simply click anywhere outside the pop-up window.

Deleting a preset files or sub-folder

The **Delete button** deletes the selected file or folder. *This happens immediately, and no “are you sure?” box will pop up,* so use this with care!

Remember, you can use the “Open in Finder/Explorer” button to open any of the preset (sub-)folders, and then you can make back-up copies if you want.

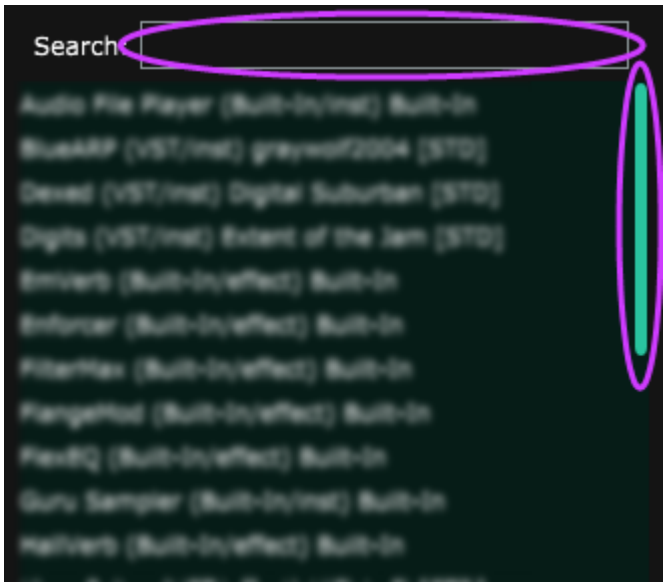
Plug-ins selector area: Finding and selecting plug-ins

The middle column of the plug-in presets view shows a big scrolling list of all the plug-ins registered in Unify's plug-in database, in alphabetical order. (If you can't find the one you're looking for, you might not have registered it; click the link for details.)

In this case, when we say “all plug-ins”, we mean ALL. Even if you select *Instruments*, *Audio Effects*, or *MIDI Effects* in the Files area, the list in the middle will still include all three kinds of plug-ins. This can be important because, for example, VST plug-ins have no way to distinguish MIDI effects from audio effects, and many VST plug-ins that are actually MIDI effects often declare themselves to be “instruments”, because many DAWs can only load them into “instrument” slots.

Finding specific plug-ins in the list

To find a specific plug-in, you can use the green *scroll bar* on the right-hand side (or your mouse's scroll function), but the fastest way is to **type any part of the plug-in's name** into the *Search box* at the top:



- As you type, the list will get shorter, showing only the entries which match what you typed
  - Don't worry about capital letters–matching is case-insensitive
- Your search text can match **any part** of the name, not just the beginning. For example:
  - Typing “VST/inst” would match all VST Instruments
  - Typing all or part of a manufacturer name will isolate that manufacturer's plug-ins
- If you make a mistake, just back-space (or use your mouse/keyboard–it's a full text-edit box)

Selecting plug-ins in the list

Once you've found the plug-in(s) you're interested in, you can click on it to **select that plug-in**, in preparation to adding it to a preset.

- Selected items are highlighted in green
- You can **de-select** an item by clicking it a second time
- You can **select more than one item** if you wish

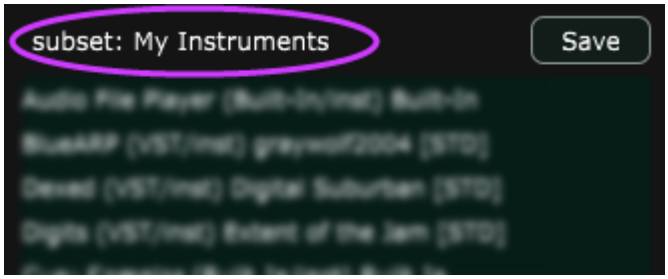
When working with a long list, it can be hard to know exactly how many items are selected. Use the “Select NONE” button at the bottom of the plug-ins selector area to make sure *nothing is selected*.

Alternatively, if your search text has narrowed down the list so that *only the plug-ins you want* are displayed, you can use the “Select ALL” button to quickly select them all.

Adding selected plug-ins to the current subset

When you have selected one or more plug-ins in the *all plug-ins area*, you can add them to the currently-selected subset by clicking the “Add selected” button at the bottom-right corner of the plug-ins selector area.

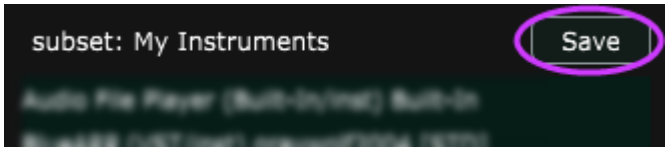
- If the “Add selected” button is “greyed-out” (indicating that it's inactive), this is because no subset is selected in the Files area. You might have de-selected it by mistake. Go back and click on the subset you want in the Files area (leftmost column).
- Whenever you select a subset in the Files area, its name will appear at the top of the Current-subset editor area (rightmost column) as shown below. If it says “Choose or create a subset”, that's another indication that you haven't selected a subset.



Editor area: saving your changes, deleting plug-ins

The Current-subset editor area (rightmost column) displays an alphabetical list of all the plug-ins in the currently-selected subset, whose name (as mentioned above) appears at the top. **If the list is empty**, it might just be empty (i.e., you haven't added any plug-ins yet), or you might have accidentally de-selected the subset in Files area (leftmost column). If it says “Choose or create a subset”, and the “Save” button is greyed-out, that's why.

Once you've finished all the editing you want to do, click the “Save” button to save your changes:



To **delete plug-ins from the subset list**, you must first **select** them. Use the mouse to **select** or **de-select** individual plug-ins, or use the “Select ALL” or “Select NONE” buttons at the bottom) to adjust your selections. Selected items will be highlighted in green. When you're ready, click the “Remove selected” button below the list. Then *make sure to click the “Save” button at the top-right* to save the change.

Problems and Solutions

This page is UNDER CONSTRUCTION and will change as users report specific issues. If an issue you're facing isn't listed here, let us know (UnifySupport@PlugInGuru.com) and we'll add it.

Known issues and limitations in Unify 1.0

As with any “version 1.0” product, Unify still has a few limitations that we'll be working hard to fix through the coming weeks and months.

Problems installing the Unify Standard Library

If you accidentally CANCEL loading the *Unify Factory Content.guru* file during installation: If you hit the “Cancel” button instead of “Open” when selecting the *Unify Factory Content.guru* file, just click the “gear” icon to go to Unify's Settings view, and **click the “Select .guru file…” button**. It's right in the middle—the only colored button on the page.

Duplicate entries in patch browser

A handful of users (all on Windows?) have reported seeing duplicate and even triplicate entries in Unify's patch browser. We don't yet know the cause, but there is a work-around, as follows:

1. Close Unify if it's open.
2. Open the main Unify data folder, and go inside the Libraries sub-folder.
    - If you see a file called *favorites.xml*, RENAME it (e.g. *OLD favorites.xml* or something).
    - Locate the file *presets.db* (this is the patch database) and drag it to the Trash.
3. Run Unify again.
4. Rebuild the patch database.

Rebuilding the patch database in this way will cause Unify to forget which patches you may have marked as favorites. The renamed *favorites.xml* file MAY be useful to re-mark them. Open it in a text editor and you MAY see a list of the patches you had marked as favorites before. (We say “MAY” because whatever underlying issue is causing the multiple patch entries could also affect the favorites file, so we can't be sure.)

Plug-in scanning issues

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 will often correct the problem.

In a few rare cases, the *PlugScanner* helper app may get stuck on one particular plug-in, re-starting again and again. This should not happen at all in Unify v1.0.6 and later, but IF it does:

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.

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@pluginguru.com if you have any information to share.

Plug-ins that Unify cannot yet load

We are working hard to track down why Unify can't load certain plug-ins. The following is a list of plug-ins which **at least some users** have reported being unable to load - many others are. The plug-in format and operating system are indicated in parentheses, where known.

- **AIR Music Technology**
  - Vacuum Pro
- **Lennar Digital**
  - Sylenth
- **Slate Digital**
  - All AU plug-ins? (Mac)
- **Softube**
  - ALL Softube plug-ins (Mac)
- **Steinberg**
  - HALion 6 (VST, Windows)
- **Toontrack**
  - Superior Drummer, EZKEys or any other Toontrack products
- **u-he**
  - Twangström (VST, Windows)
- **UVI**
  - Falcon (VST, Windows 10)
- **Waves Audio**
  - All instrument plug-ins

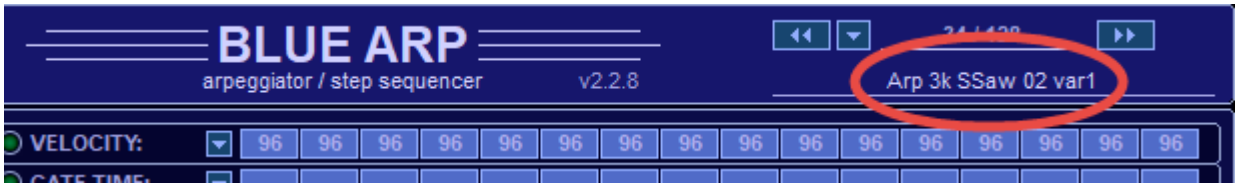
- SOME audio-effect plug-ins

Please use the PlugInGuru Forums to let us know about any more you may discover. We have set up separate forums for plug-in issues on Windows and Mac computers, because it often happens that a plug-in may work on one, but not the other.

ALSO - another alternative way to load Plug-Ins is to drag the plug-ins from your finder into Unify's “Known Plugins” window (click the PlugIn icon bottom Right cluster). This bypasses the scanner and will allow you to load as many files as you select in your finder and drag into Unify,

Issues with bundled plug-ins

BlueARP: Clicking on pattern-name causes crashing on Windows systems only. Does not happen on Macintosh. This appears to be an issue in BlueARP itself, and we will work with the author to try to resolve it.



Having your say

We very much want to hear your feedback about your experience with Unify–what works for you and what does not. The very best way to talk about this is to register at our new PlugInGuru Forums web site. Forum discussions can be seen by all members, so by posting your issues there, you'll be helping other Unify users who may experience similar issues.

For any communications you prefer to keep private, contact us directly at UnifySupport@PlugInGuru.com.

Tour of the layer-stack view

You'll spend most of your time in Unify's *layer-stack view*, which you can always get to using one of the two icons at the rightmost end of the icon strip:

Scrolling vs. Accordion view

- The icon on the left shows you the *scrolling view*, where all of the layers (except possibly MIDI layers) will be shown full-size, and if they don't all fit, a green scroll bar will appear on the right-hand side.
    - You can scroll up and down by dragging the green scroll bar itself, OR
    - You can use your mouse's scroll wheel (or ball, or touch surface, etc.) anytime the mouse pointer is over the layer stack.
- The icon on the right shows you the *accordion view*, where you'll see ALL layers in the stack, even if they have to be squashed vertically to fit.

The following pairs of screenshot show how the same factory patch (*BPM SPLIT - And So It Begins!*) appear in the scrolling (left) and accordion (right) views:

Note that you can also **resize the Unify window** to see more (or fewer) layers.

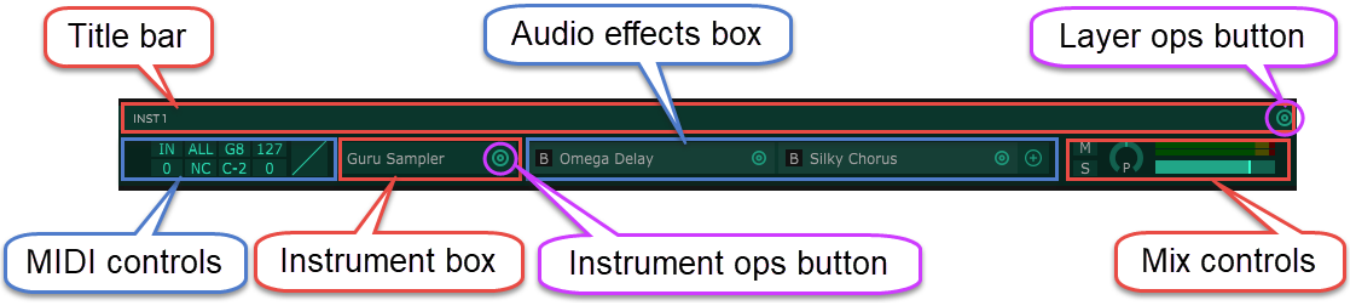
Showing/hiding MIDI layers

Because you often don't need to see the details of MIDI layers after they are fully set up, Unify provides the option to “show” or “hide” them. When you click the **Layer button** just below the Unify logo (in the header part of the GUI), you'll see the *show/hide MIDI layers* item at the bottom of the menu.

Instrument layers

Instrument layers are the most important type, and also the most complex, in that it contains elements that other layers don't. Once you understand the structure of instrument layers, you'll understand the others as well.

Each Instrument layer is divided into five rectangular regions as shown:



1. The **title bar** contains a *title prefix* (“e.g. INST 1”) and may also contain a *layer title* which you specify yourself.
    - Double-click the layer title to turn it into an editor for the title. Press Enter/Return on your keyboard when you are finished editing.
2. The **MIDI controls** area contains a dense cluster of MIDI-related controls, described in detail below.
3. The **Instrument box** contains the name of the instrument plug-in for the layer.
    - Double-click the name to open the GUI window for the plug-in
4. The **Audio effects box** contains the name of any audio-effect plug-ins used on the layer, together with some related controls, described in detail below.
5. The **Mix controls** area contains several controls related to how this layer's output is mixed in Unify, described in detail below.

Ops (operations) buttons

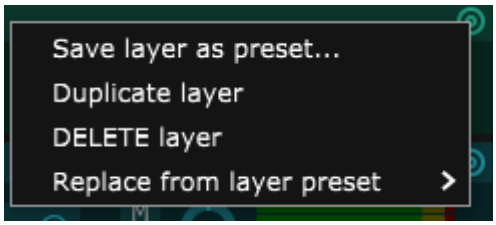
Any time you see an icon consisting of two concentric circles in Unify, it's called an “ops button” or “operations button”, and clicking it will pop open an “ops (operations) menu”. The scope of the operations (functional items) on the ops menu always matches the location of the ops button in the layout.

For example, at the extreme right-hand end of the title bar is the **layer ops button**. Its location in the layer title bar is as a graphical hint to indicate that the operations its associated menu provides are related to the entire layer. Similarly, the **instrument box** features an **instrument ops button**

for operations on that instrument plug-in, and as you can see, a similar ops button exists for every audio-effect plug-in.

Layer ops button and Layer presets

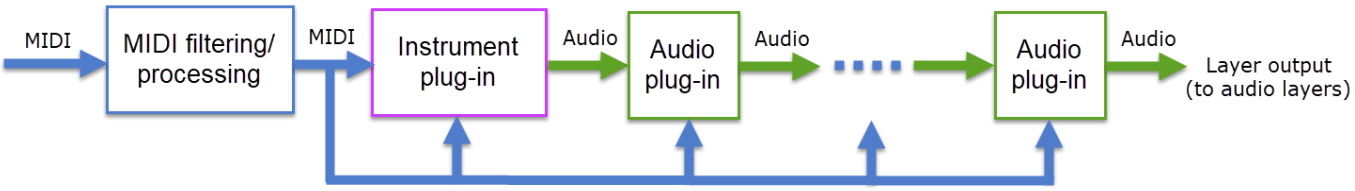
The Layer operations menu for an Instrument layer looks like this:



- The “Save layer as preset…” option allows you to define a new **layer preset** which saves the entire state of the layer, including the state of all its plug-ins, mix settings, etc.
- “Duplicate layer” creates a new layer with the identical state (without having to save a layer preset)
- “DELETE layer” removes the layer
    - NOTE deletion is immediate; there is no “are you sure?”
    - Option-click (Alt-click on Windows) on the **layer ops button** is a shortcut, immediately deleting the layer
- “Replace from layer preset…” allows you to replace the entire state of the layer based on a layer preset you created previously.
    - NOTE the mix parameters (pan and level) are NOT updated when replacing from a layer preset

Signal flow in instrument layers

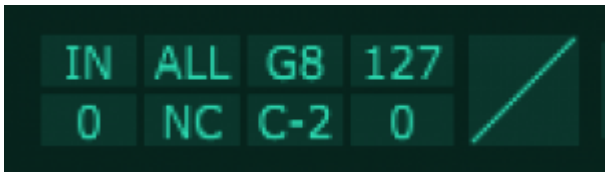
The horizontal layout of the MIDI, Instrument, Audio-effects, and Mix controls sections reflects the signal-flow in each layer, with MIDI coming in on the left, and audio going out on the right.



- The incoming MIDI stream is filtered/modified according to the MIDI controls, then passed on to ALL plug-ins.
- The instrument plug-in responds to MIDI messages by generating audio output, which are passed to the audio effects chain
- All the plug-ins are connected in daisy-chain fashion–the output of the instrument plug-in feeds the input of the first audio effect, whose output feeds the second, and so on.
- The output of the last plug-in in the chain (which would be the instrument output, if no audio effects are used) is the output of the entire layer, which is mixed to the audio layers (see below) according to the settings of the mix controls.

See signal flow for a diagram and discussion of overall signal flow in Unify.

MIDI controls

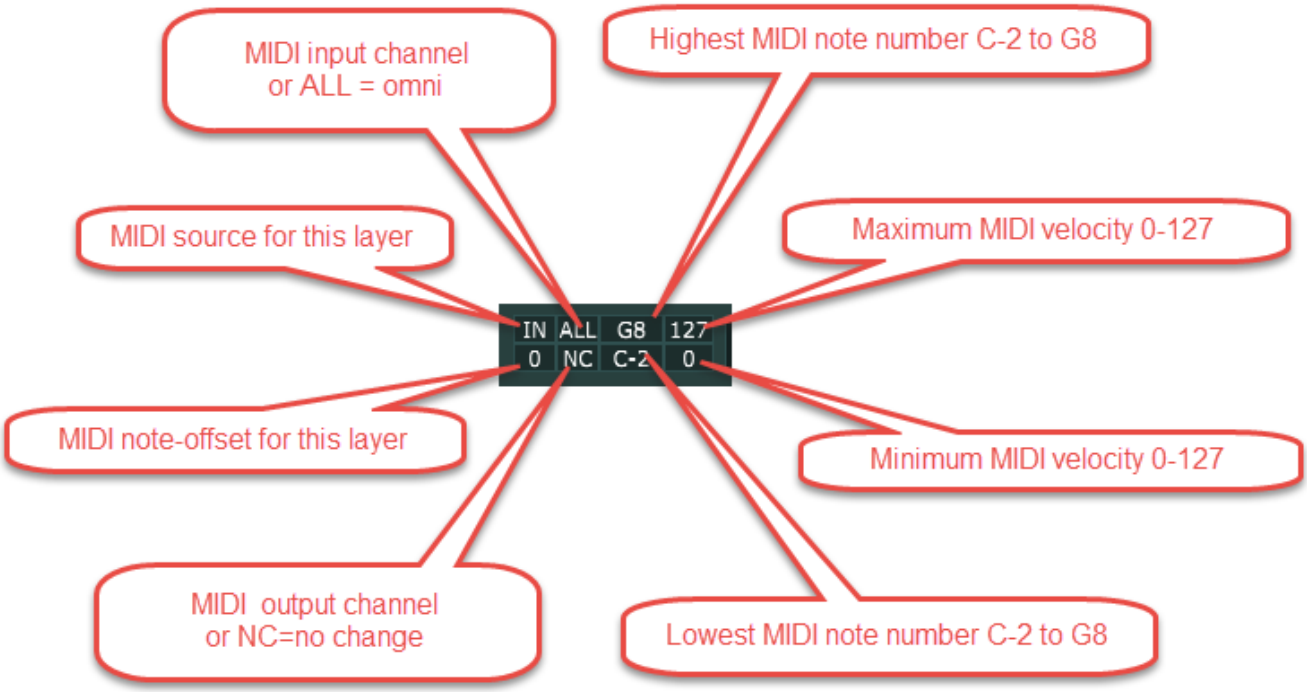


The stream of MIDI data coming in to each instrument layer is subject to three kinds of operations, all set using the cluster of MIDI-related controls at the left-hand side of the layer (see diagram below):

1. **Stream selection**: You can select whether the incoming MIDI comes direct 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

See How Unify processes MIDI for more details.

The following eight MIDI controls are always visible:



In addition, two more items may be visible in the blank area just to the left of the main cluster of eight MIDI controls:

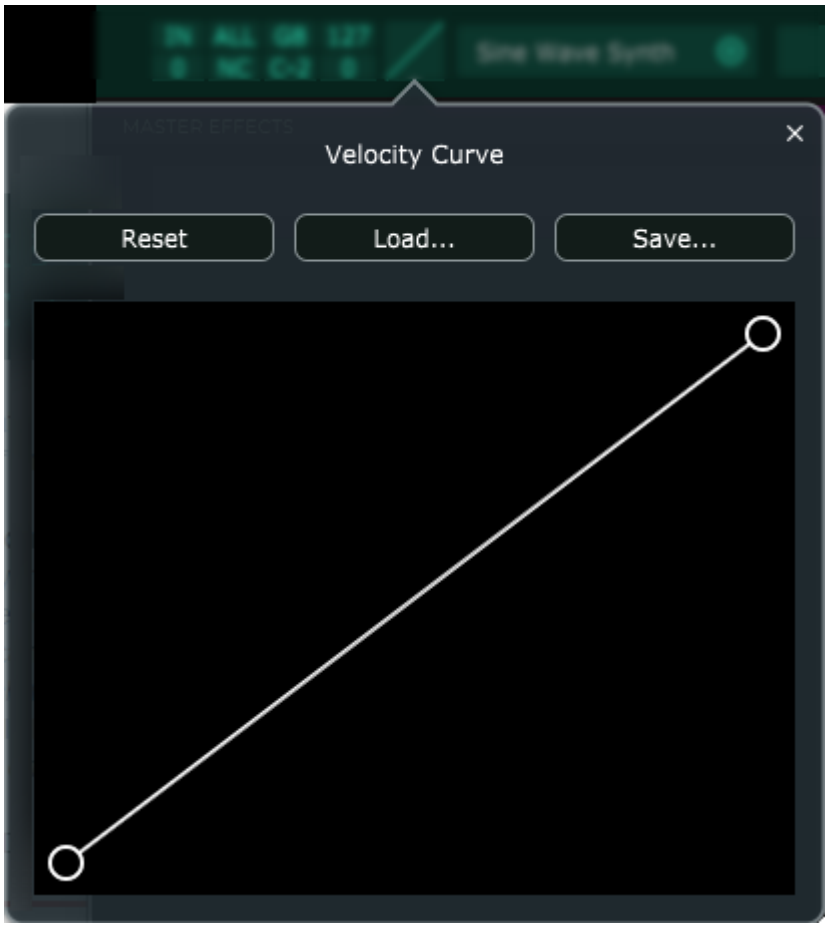
1. **MIDI activity light**: a green dot will flash to indicate that MIDI events are coming into the layer from the currently-selected MIDI stream.
2. **Latch button**: When the selected MIDI stream is Unify's main MIDI input stream, you can optionally choose **mono** or **poly latch** modes (described below). When either latch mode is selected, a ninth rectangular button will appear to the left of the **MIDI note-offset box**, which you can click to toggle latching on/off.

All the rectangle-shaped controls can be controlled with either the *left* or the *right* mouse button (use Ctrl+left on a Mac with a single-button mouse):

- The five controls that show a *numeric value* or a *note name*:
  - can be adjusted by a *left click* followed by a *drag*, OR
  - *right-click* to pop up a small *edit window* where you can use your mouse and keyboard to edit
  - MIDI note-number pop-ups are special, in that you can also *press a MIDI key* to set the value
- For the other three controls, both *left*- and *right-click* pops up a *menu* of available choices
- The ninth *latch button* which appears in latched input modes is a simple on/off toggle

Velocity-curve editor

Clicking on the **MIDI Velocity Graph control** at the right-hand side of the MIDI-controls cluster pops up a *curve editor window* like this:



The graph itself represents *incoming* MIDI note-velocity values along the horizontal axis, with lowest (quietest) velocities on the left and highest (loudest) velocities on the right. The vertical axis represents *outgoing* (processed) velocity values, lowest at the bottom and highest at the top. The default velocity curve is a straight line–the so-called “identity mapping” where each incoming velocity value is mapped to the identical value (no change at all).

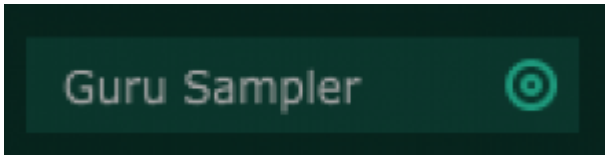
- The **Reset button** reverts the curve shape back to the straight-line “identity” curve, as shown
- The **Load… button** lets you load a previously-saved curve-shape preset (XML file)
- The **Save… button** lets you save the current shape as a new curve-shape preset

In the graph control itself, you can:

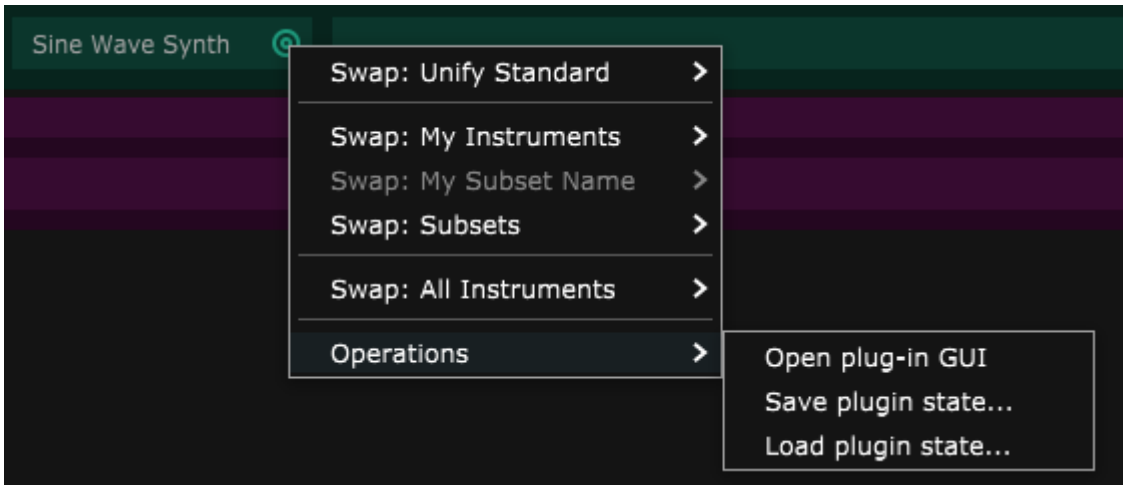
- Click and drag the *endpoints* (circles) up and down
- Click anywhere *between* endpoints and drag up/down to change the curvature of that segment of the curve
- Double-click anywhere *between* endpoints to *create a new split point*, splitting the segment into two
- Single-click directly *on a split point* and drag to move it up, down, left, or right.
- Double-click directly *on a split point* to *delete it*, joining two segments back into one

For velocity curves, you will rarely need to create split points. You will usually only need to adjust the curvature (*slightly* – a little change goes a long way). Dragging upward to create a *convex* curve will make the layer *more responsive to velocity*, and can be useful when you are working with a weighted-key MIDI controller. Dragging downward to create a *concave* curve will make the layer *less responsive to velocity*, and can be useful when playing a very lightweight synth-action keyboard.

Instrument box



The Instrument box is simple. The name of the current instrument plug-in is displayed, and you can double-click on it to open that plug-in's own GUI window. At the right-hand end is the *instrument ops button* (icon), which you can click to pop up the *instrument ops menu*:

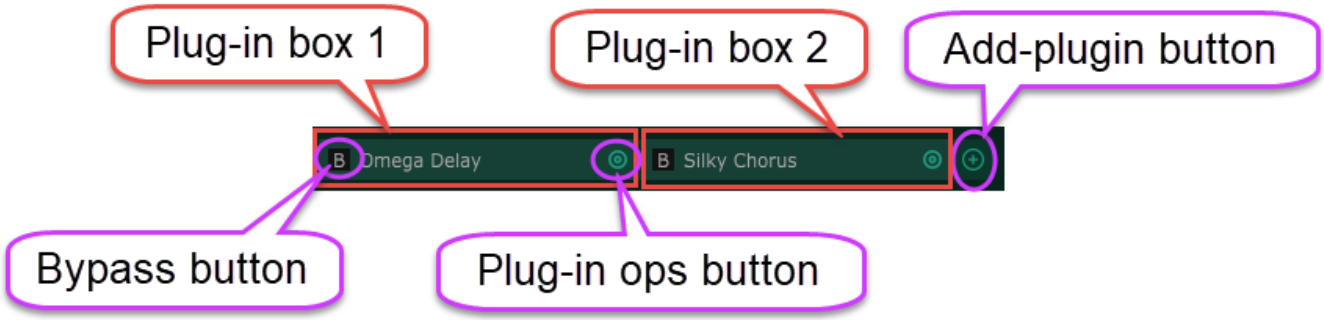


The instrument-ops menu has several sub-menus, and is divided into four sections:

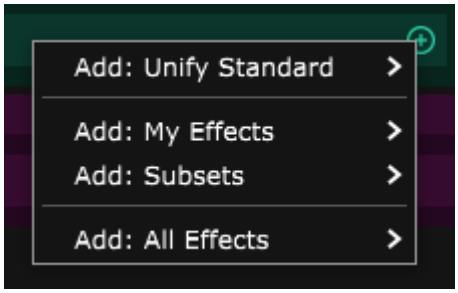
1. At the top is the “Swap: Unify Standard” sub-menu, which allows you to replace the current plug-in with any of Unify's built-in or bundled plug-ins.
2. In the middle, any number of “Swap” sub-menu items might appear, based on whatever plug-in subsets you have defined.
3. The “Swap: All Instruments” sub-menu is generated automatically, to ensure that ALL known plug-ins (see Scanning and using your own plug-ins) are listed, arranged hierarchically by manufacturer name
4. At the bottom, the “Operations” sub-menu provides a single-click alternative for opening the

plug-in GUI, plus options to save the and load the plug-in's current state as an XML preset.

Audio effects box



The audio-effects box on each instrument layer is initially empty, except for the *add-plugin button* at the right-hand side, which pops up a menu very similar to the “Swap” items on instrument-ops menu:

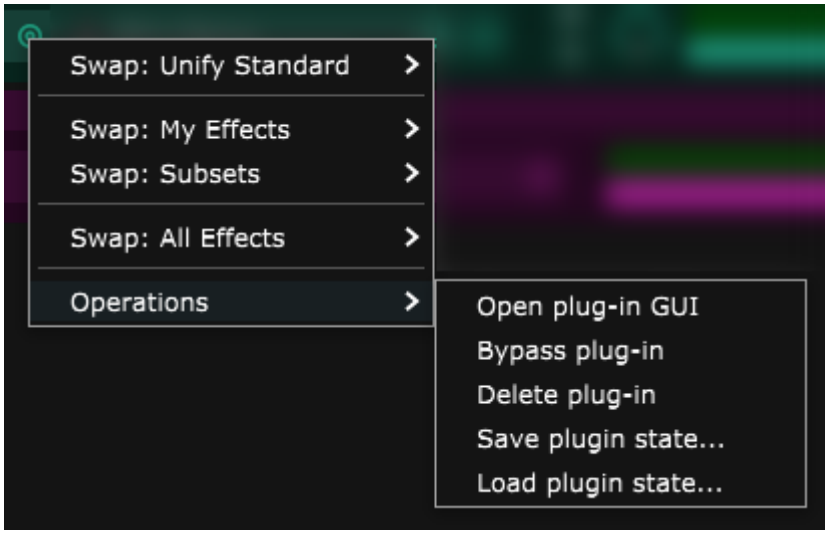


As with the “Swap” menus, this is divided into three sections:

1. Add: Unify Standard, for built-in and bundled plug-ins
2. Add based on plug-in subsets
3. Add: All Effects, for a hierarchical list of all plug-ins Unify knows about on your system

When you add an audio-effect plug-in using this menu, a new **plug-in box** will be added at the right-hand side, indicating that it follows the previous ones in the audio processing chain. Each plug-in box is almost exactly like the intrument box described above, including the presence of an *ops button* for operations specific to that plug-in, but it also features a **bypass button** on the left, which you can use to toggle whether the plug-in is active in the processing chain (normal case) or if it is *bypassed*.

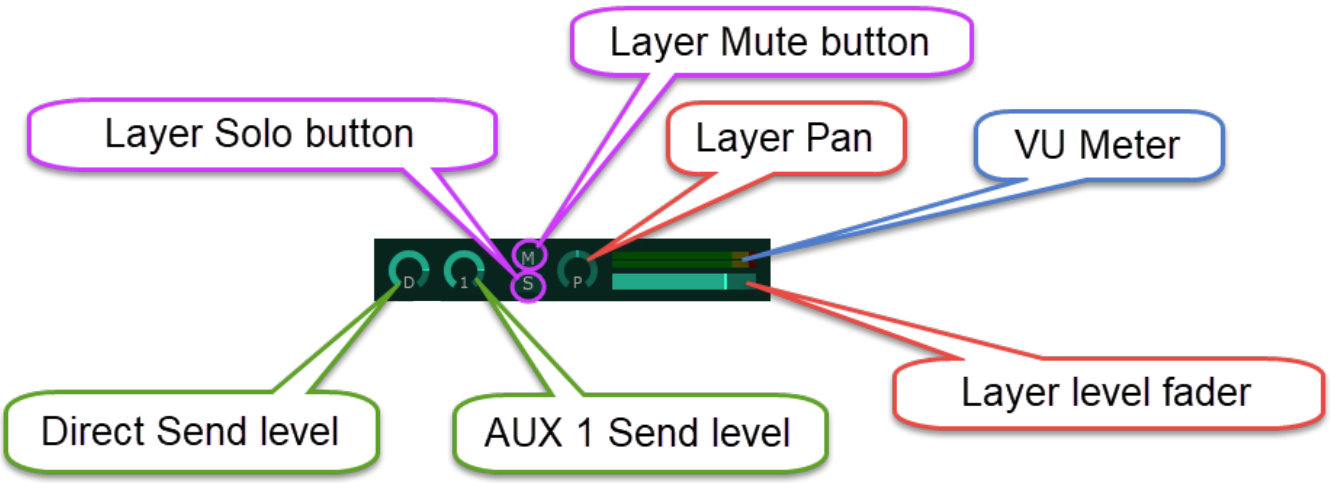
The *effect ops-menu* also offers slightly different options, including *bypass* and *delete* functions:



Finally, note that you can *drag the individual effects boxes* to *change the order of the effects chain*. Red markers will appear as you drag, indicating where the effect will be moved to when you release:

Mix controls

The *mix controls* cluster at the right-hand side of each instrument layer has a dense array of controls:



The **Direct** and **AUX Send level** controls only appear if the patch has at least one AUX layer. These allow you to control how much of the layer's output is sent *directly to the Master Effects layer* and how much is sent to each individual (numbered) AUX layer. Note these are “post fader” levels, i.e., they are affected by the settings of the main **layer level fader** (and also the pan control).

The **Layer Mute** and **Layer Solo** buttons are straightforward, and work as in any DAW:

- use the mouse to toggle muting and soloing on/off
- you can Solo more than one layer at once
- *right-click* (or Ctrl+click) a Solo button to solo that layer alone (any others will be un-soloed)

The **Layer Pan** control and **Layer level fader** are also straightforward and similar to those in DAWs,

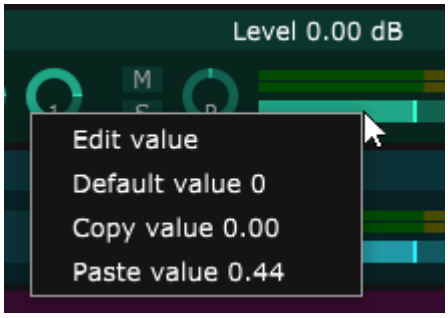
as are the *VU Meters* above the fader.

- The top half of the VU meter is for the Left channel; the bottom half is for the the Right
- VU meter range is from -60 dB to 0 dB; -10 dB to -3 dB range is yellow, -3 dB to 0 dB is red
- Signals above 0 dB will *almost certainly be distorted* – keep them out of the red range

Features common to level, pan, and fader controls

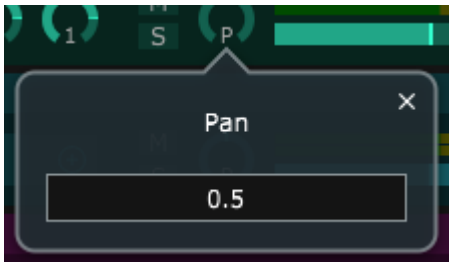
All faders, knobs and knob-like controls in Unify have certain common characteristics:

- Rolling the mouse pointer over the control will usually result in its exact value being displayed nearby
- Adjust value by clicking and dragging: downward/leftward to reduce, upward/rightward to increase
- Double-click resets value to the control's default
- Right-click to pop up a *value-edit menu*:



- “Edit value” pops up a small window where you can type a numeric value (see below)
- “Default value” restores the control's default value (same as double-clicking)
- “Copy value” copies the numeric value to the clipboard
- “Paste value” pastes the value previously saved to the clipboard
- Note edited/pasted values are automatically restricted to the valid range for the control

Pop-up value editors



In all pop-up “Edit value” windows:

- you must click the value with the mouse before you can start typing
- hit the Enter/Return key on the keyboard to save the edited value, OR
- click the small “X” at the top-right corner (or simply click anywhere outside the box) to close the edit box without changing the value.

Creating your first patch (Tutorial)

On this page, we'll work through a simple example of creating and saving a basic piano-and-synth patch. Because some of you will be reading this when you aren't familiar with Unify yet, this page will take a step-by-step, *tutorial* approach, unlike the rest of the manual pages, which are more *encyclopedic*.

You'll be working almost entirely in Unify's layer-stack view You needn't click that 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.

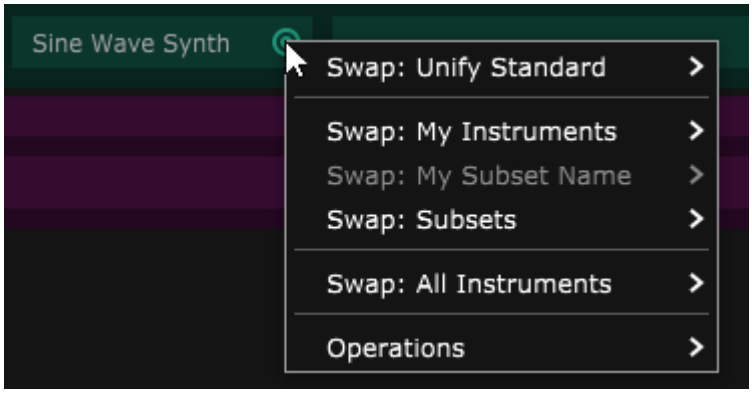
Start from an INIT patch

To start creating a new patch “from scratch”, click the “init” button at the right side of the header, just under the “browse” button. You will then see the simplest possible layer stack in Unify, with just a single *instrument layer* and an empty *Master effects layer*. Because an instrument layer must always have an instrument plug-in loaded, the built-in Sine Wave Synth is used, and there are no effects.

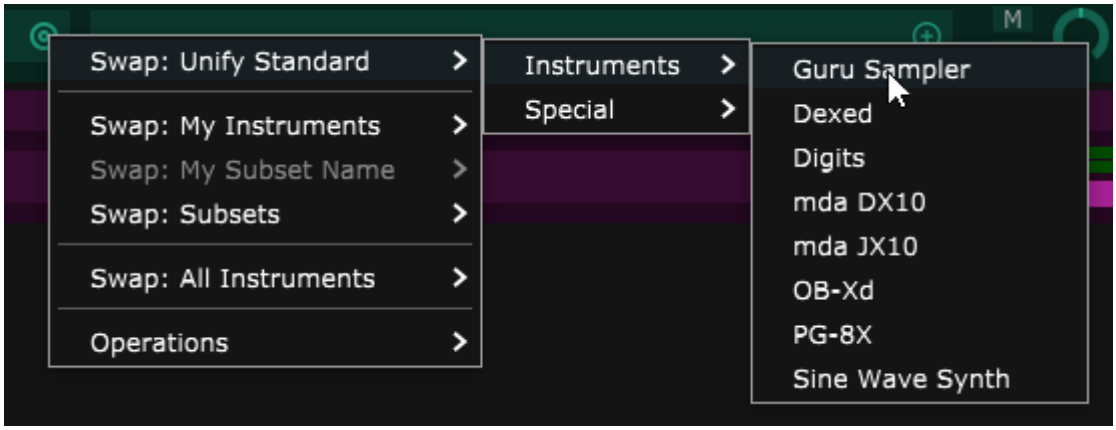
Play a few notes on your MIDI keyboard to make certain Unify is working correctly. If you don't hear any sound, go back and check that you have correctly set up your MIDI and audio settings.

Create the piano layer

Click the *ops button* (two concentric circles icon) to the right of the words *Sine Wave Synth* to pop up the *instrument ops (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”:



The plug-in name will change from “Sine Wave Synth” to “Guru Sampler”, and the Guru Sampler GUI window will open. Don't worry about the details at this point; just look for the row of three pop-up menus about one-third of the way down from the top. Click the middle menu to select “Keyboard”, and select “HR Pop Piano f” on the right menu, 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* GUI window by clicking on the small red “X” in the title bar. NOTE the “X” will be *on the right* on a Windows system (as shown above), but *on the left* on a Mac.

Add an audio effect

Click on the **add button** (plus sign in a circle) at the right-hand end of the **effects box** as shown below:

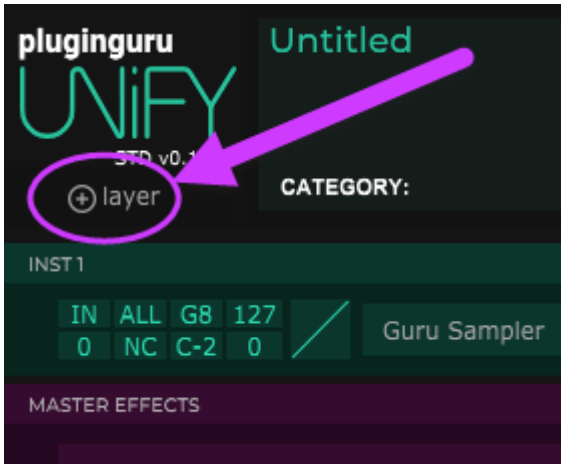
In the menu which pops up, follow “Unify Standard” through “Reverb” to “EmVerb” and click on that:

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

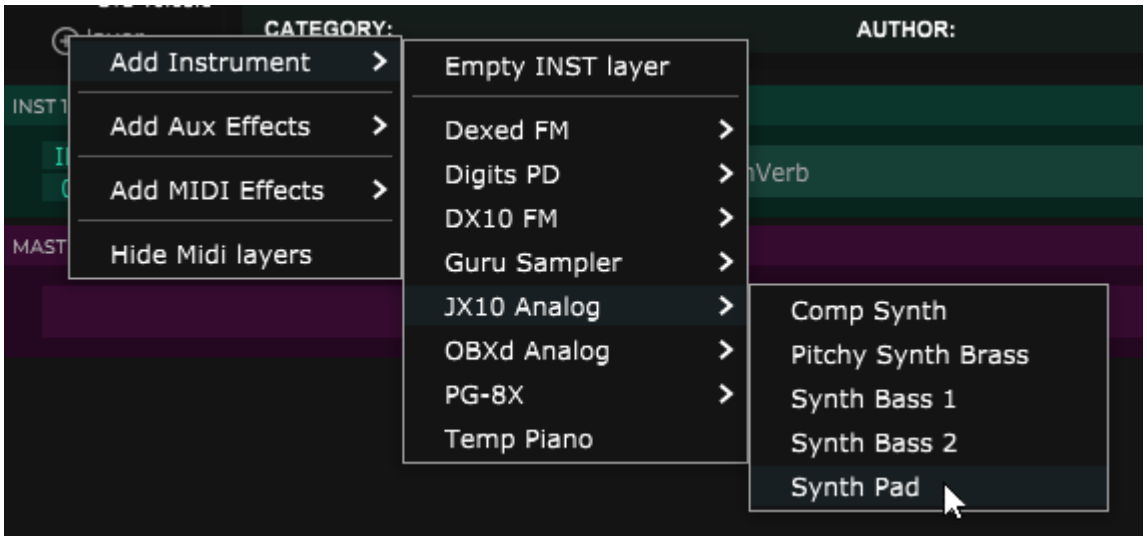
Close the *EmVerb* GUI window by clicking on the small red “X” in the title bar.

Add the synth layer

Locate the **add layer button** just below the Unify logo at the top-left corner of the Unify GUI.



Click on that to open the *add-layer menu*, which is another system of cascading sub-menus, but this one lets you select fully-configured *layer presets*. Follow “Add Instrument” > “JX10 Analog” > “Synth Pad” to add a new instrument layer using an instance of the *mda JX10* synthesizer:

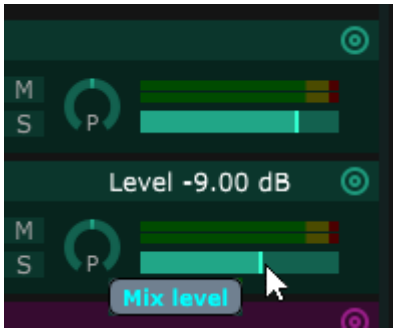


The GUI window for *mda JX10* will open, but at this point, we don't need to make any changes, so just close it again.

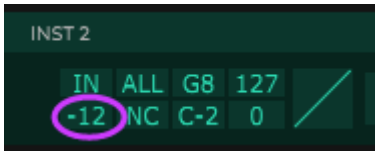
Play a few notes on your MIDI keyboard to confirm that you hear piano and synth tones layered together.

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, locate the **MIDI note-offset control** at the bottom-left corner of the *MIDI-controls cluster* for layer 2, where it says “-12”:

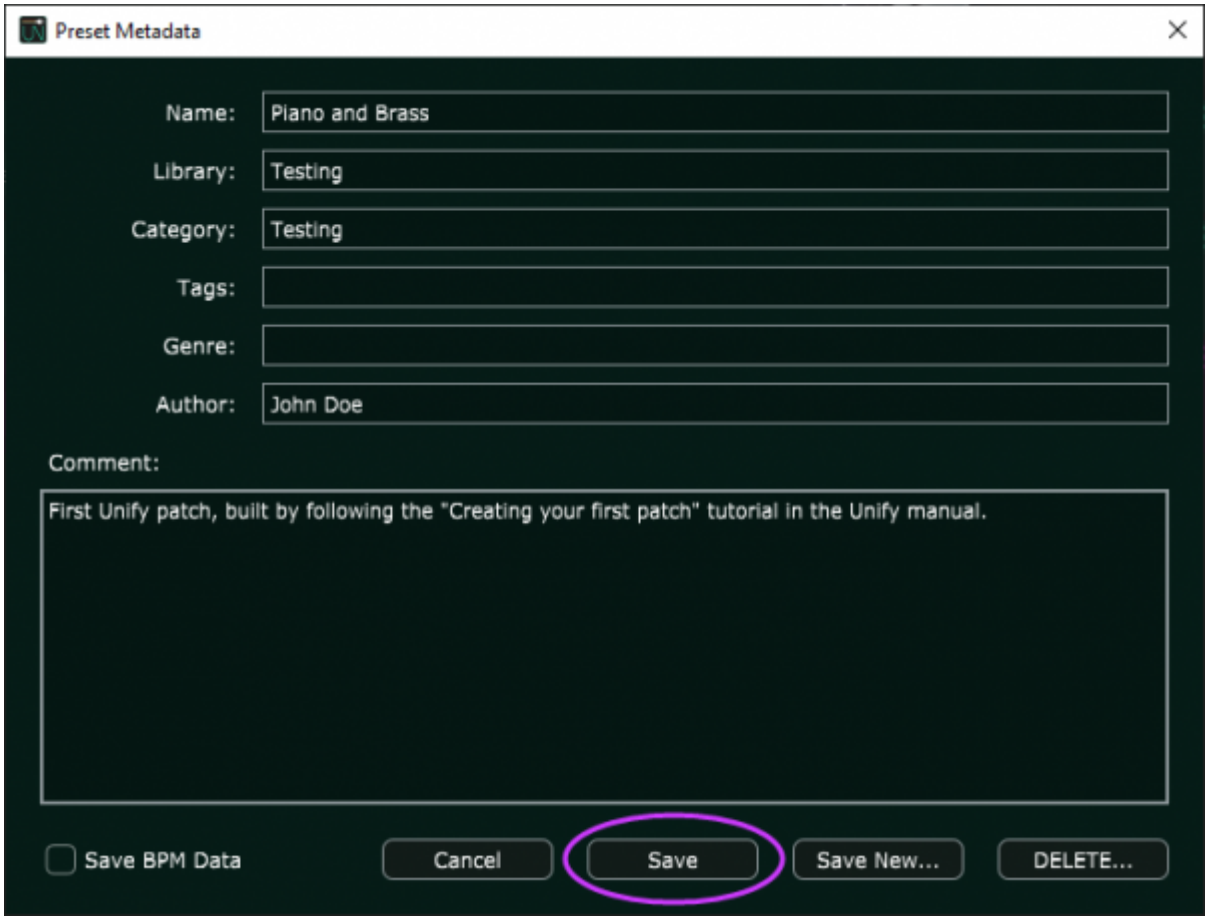


Roll your mouse pointer over the control, press the mouse button, *drag upwards* (keeping the mouse button down) until the value reads “0”, then release. Play some notes now, to verify that the two layers are now sounding in the same octave.

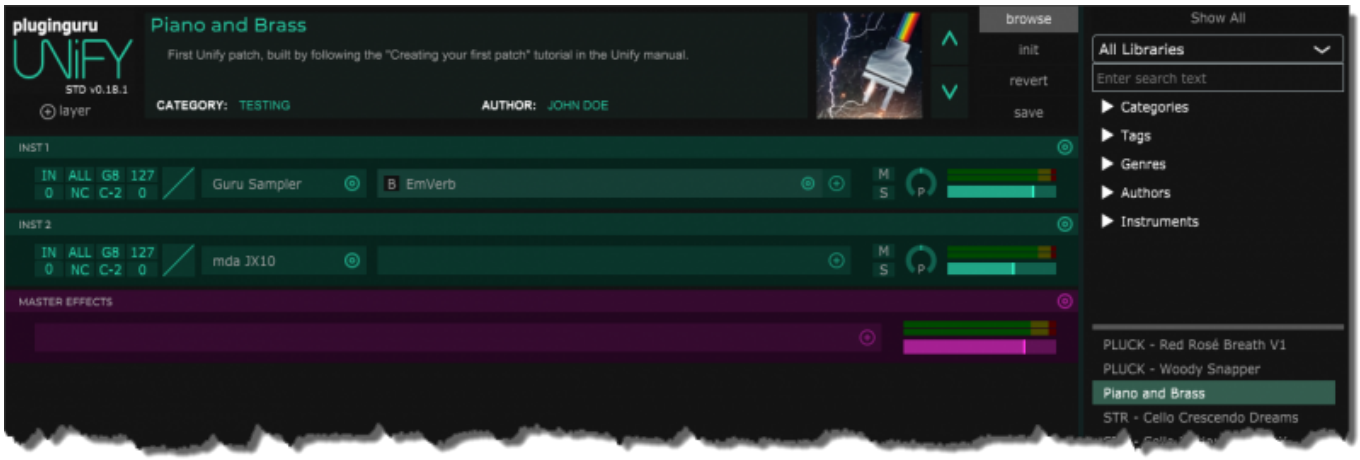
Save your new patch

At this point, you should SAVE your new patch. Locate and click the **save button** in the upper right-hand corner of the Unify GUI:

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” button.



When you click the “Save” button, you'll be presented with a standard file-save dialog, which will be different for Mac vs. PC, but should be familiar to you. Notice that, because you entered “Testing” as the Library name, Unify has already created a new folder called “Testing” and a sub-folder called “Patches” inside it, and is prepared to save your new patch file there. Also, the *file name* field has already been filled in for you, based on the name you gave to the patch (e.g. “Piano and Brass.unify”). Click the “Save” 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 GUI, 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!

Make some more changes

Now that you have saved your patch, you can make all sorts of additional changes. If you're happy with your changes, re-save the patch. If you're no, click the **revert** button just above the **save** button to re-load the last saved version.

Changes you might want to make at this point include:

- Double-click each layer title to add your own descriptions
    - Remember to hit Enter/Return on your keyboard to confirm
- Add instances of *Omega Delay* and *Silky Chorus* on layer 2
- Use the **layer ops button** at the right-hand end of the Master Effects layer, choose “Replace from layer preset” > “Unify Master presets” > “MASTER Optimizer JL” to put John Lehmkuhl's pre-built mastering effects chain on the combined output.

When you hit the **save** button to save the patch, and then the “Save” button at the bottom of the “preset metadata” dialog, the previous preset file will be overwritten immediately. If you'd like to save a new file, **change the patch name** and then save. (You could also click “Save As” and change only the file name, but that will be confusing, as you would then end up with two identically-named patches in the patch list.)

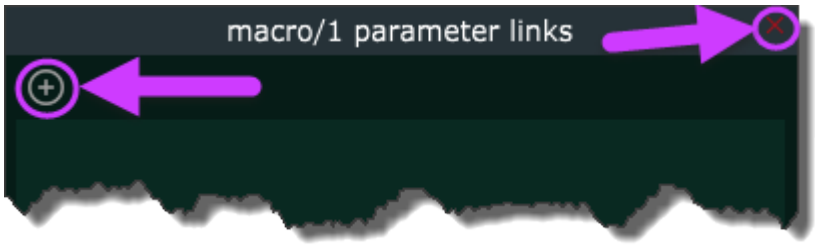
Use a macro to adjust the layer balance

True *PlugInGuru* fans know that Skippy never makes a patch without making sure at least *something* happens when you use the mod-wheel, so let's do that now.

Click the **knob icon** at the bottom-left in the icon strip, to open the Macro knobs view in the footer:

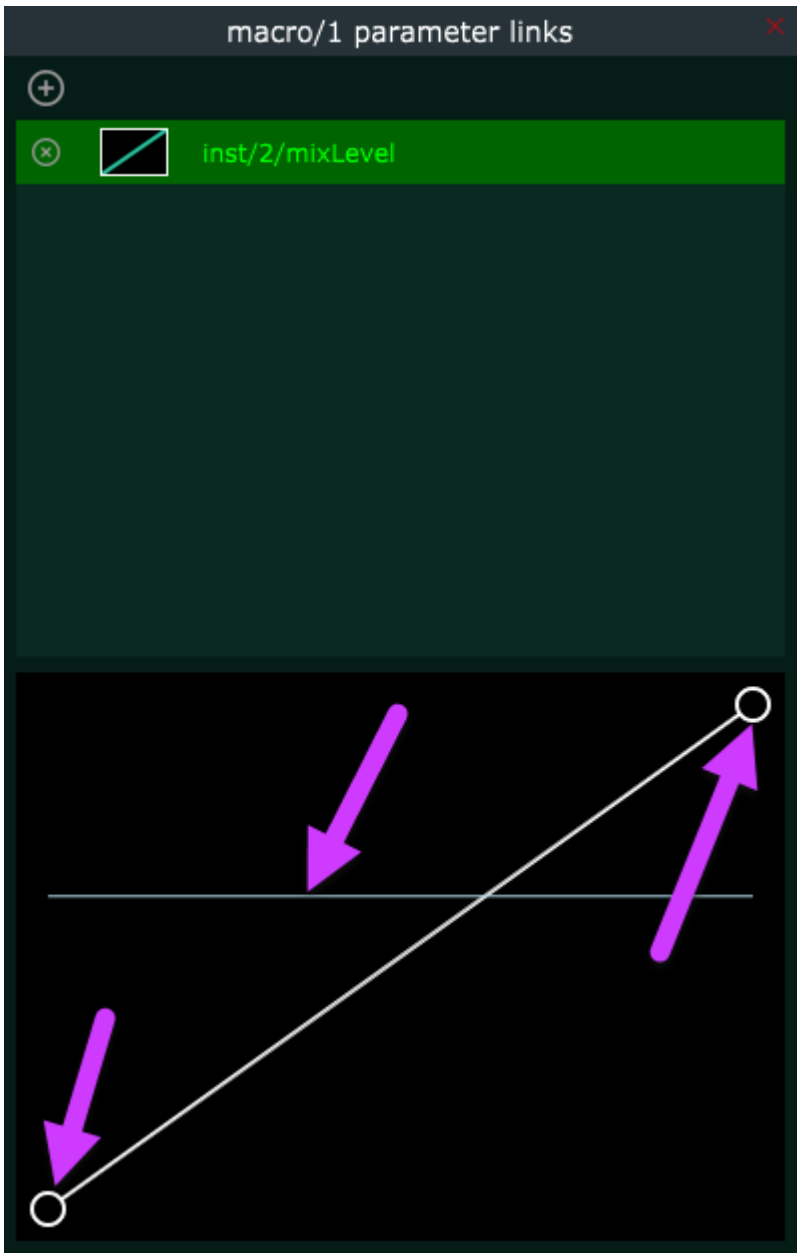
If the first knob on the left doesn't say “CC1” in the middle, click the **ops button** just above and to the right of the knob, and choose “Assign CC” > “CC1 Modulation Wheel (coarse)”.

We'll connect the *Layer 2 level-fader* parameter to the mod wheel. Click the first knob's ops button, and choose “Linked Parameters…” to open the **linked-parameters editor** window. When this first appears, it will be empty, except for the title, as shown:



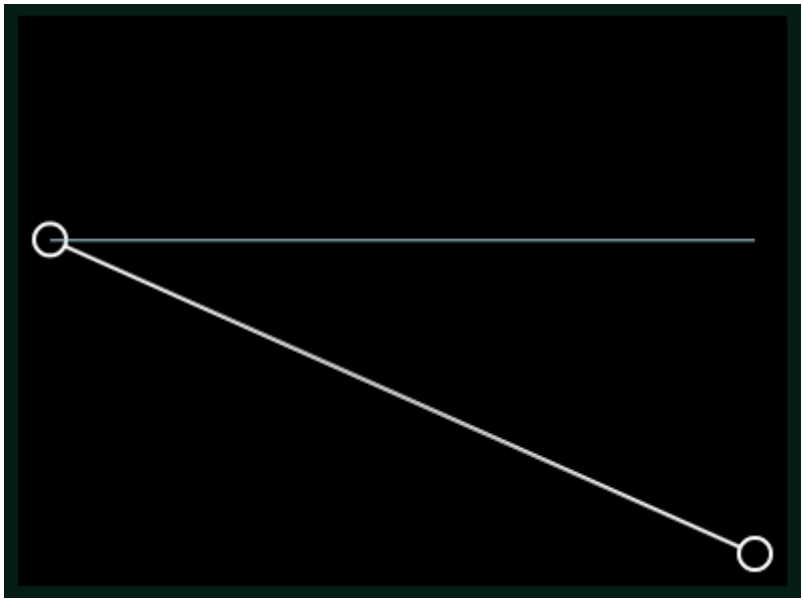
Note the small red “X” in the title bar (on the *left* on Mac, on the *right* on PC); you'll use this to close the window in a moment. Click the **add parameter button** (small plus-sign in a circle) to open the usual cascade of menus, and choose “inst” > “2:” > “mixLevel”.

The view should change as follows. In the **response-curve editor** section at the bottom, note the two circular *endpoints* and the *horizontal blue line*.



The blue line represents the current setting of the Layer-2 *mixLevel* parameter. We'd like the patch to start out with Layer 2 at this level, and fade down to nothing as the mod wheel is raised. Drag the two

endpoints so the first one aligns with the blue line and the second is all the way down, like this:



Close the linked-parameters window by clicking the red “X” in the title bar, and try playing with your MIDI controller's mod-wheel while playing some notes. As you bring the mod wheel up, you should hear the synth layer fade away until eventually all you hear is piano.

Quick–**save that patch again**–and don't forget to add a note to the description, e.g. “MW: fade out synth”. Congratulations again–you've made a proper PlugInGuru patch, complete with mod-wheel support!

Learning more

At this point, you should be starting to get the feel of the Unify GUI, enough that you shouldn't need further hand-holding. The other pages in this manual will thus be less “tutorial” and more “encyclopedic”.

- See Basic patch recipes for guides to making very common layers and splits
- See Tour of the layer-stack view for a complete guide to the layer-stack view
- See Using Macro Knobs for more details about using the macro knobs and MIDI CCs

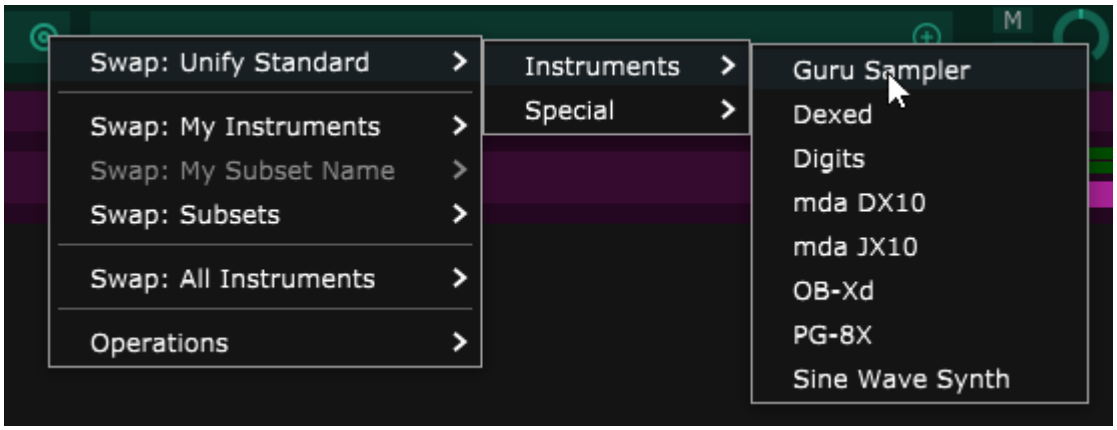
Basic patch recipes

This page gives very brief guides to setting up common patch structures in Unify.

Multiple Layers and Octaves

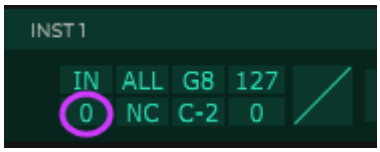
To **layer multiple sounds**, just create a new Instrument layer for each sound. By default, incoming MIDI gets routed to all instrument layers in parallel, so all layers will sound together as you play.

The quickest way to add instrument layers is to use the pre-defined *instrument-layer presets* available by clicking the **add-layer button** just below the Unify logo:



We suggest you try out all these pre-defined layer presets, to become familiar with what's available. In time, you may want to create your own layer presets–use the **layer ops button** at the extreme right-hand end of the layer title bar.

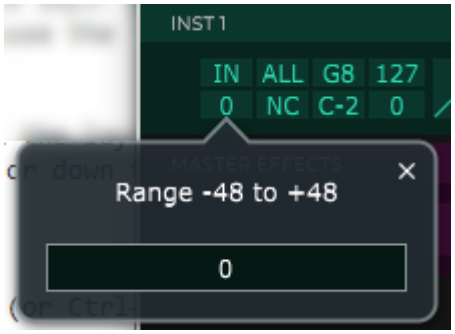
To **change the octave for a layer**, click the layer's **MIDI pitch-offset** control at the bottom-left corner of the **MIDI controls cluster**, and drag up or down to change the displayed number.



- 12 semitones is one octave UP
- -12 semitones is one octave DOWN

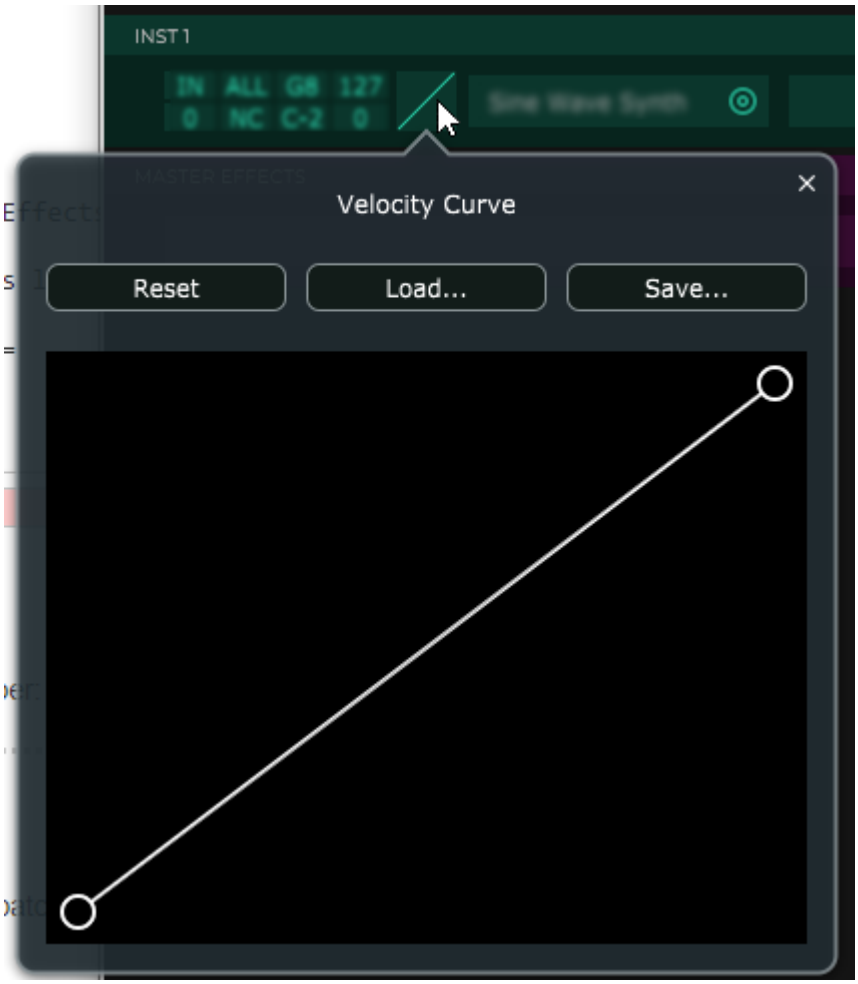
You may find it quicker to *right-click* (or Ctrl+click) the MIDI pitch-offset control, to reveal a pop-up editor box, where you can simply type in the value you want.

- You MUST click once on the displayed value, before you can type
- Hit the Enter/Return key on your keyboard when you are done.



Adjusting velocity response

To adjust the way a layer responds to MIDI key velocities, click the layer's **velocity-response** control to open the pop-up **velocity response editor**:



Click anywhere in middle part of the graph and drag UP or DOWN to change the curvature.

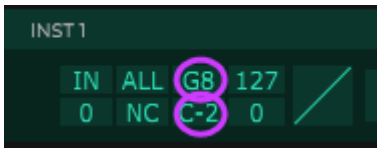
- Drag UP to make the patch sound louder (*less responsive* to velocity)
- Drag DOWN to make the patch quieter (*more responsive* to velocity)
- SLIGHT changes are best–a little goes a long way
- See velocity-curve_editor for more details

Key Split

A *key split* is two or more Unify layers configured with non-overlapping MIDI key ranges, e.g. a bass patch in the left hand and an electric piano patch in the right.

Add the layers as described above for layers and octaves.

To change the MIDI key-range for the layers, use the two **MIDI note-number** controls as shown:



Use the upper control to adjust the upper key limit for your left-hand layer:

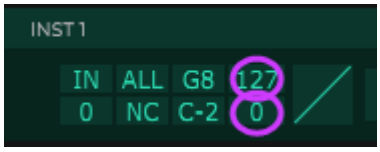
- Click on the value (default “G8”, MIDI key number 127) and *drag down* to reduce, OR
- Right-click (or Ctrl+click) to pop up a value editor
    - You can enter *either* a MIDI note-number OR a key name (e.g. C3 for Middle C*), BUT
    - While the pop-up editor is displayed, you can simply press a key on your MIDI keyboard

* Whether Middle C is called “C3” (Yamaha style) or “C4” (Roland style) is a setting available on the Settings view. Yamaha style is the default.

It's up to you to ensure that layer key-ranges don't overlap or have “dead space” in between them. You'll notice a change in the color of the layers' title-bars, which can be helpful for this. The active part of a layer's key-range is denoted with a brighter shade of the background color.

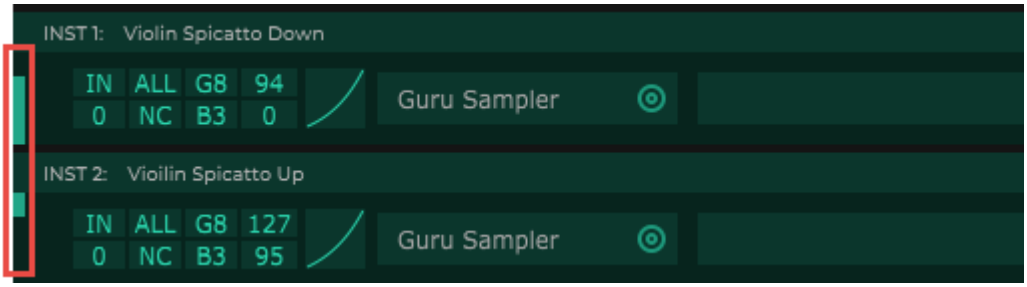
Velocity Split

A *velocity split* is two or more layers, set to respond to non-overlapping ranges of MIDI key velocity.



Setting up a velocity split is almost the same as key-split, with these exceptions:

- Use the **max-velocity** and **min-velocity** controls instead of the pitch controls (screenshot above)
- Either click/drag the controls to change the displayed velocity-value (0 to 127), or right-click (Ctrl+click) to edit the value directly; *there is no option to enter values from your MIDI keyboard*
- The layer velocity ranges are indicated using small vertical color bars on the left-hand side of each layer, rather than color changes in the title bars, as shown below.



Compression and Limiting on the Master Effects layer

You would not typically use the Master Effects layer of a patch for creative audio effects like chorus or delay, but this is an ideal place to put “mastering”-type effects such as a *limiter*, *equalizer*, and possibly a *compressor*.

The quickest way to set up your Master Effects layer is to use the **layer ops button** on the right, and select “Replace layer from preset” > “Unify Master Presets” > “MASTER Optimizer JL”. This immediately gives you Skippy's standard mastering effects chain, consisting of

1. An instance of FlexEQ which allows you to
    - sweeten the low-end, e.g. with a slight boost
    - sweeten the high-end, either up or down, and
    - use one or both of the *peaking filters* to remove any harsh or strident peaks in the spectrum
2. An instance of the Enforcer compressor, to reduce dynamic range and help quieter notes come through
3. An instance of the LoudMAX limiter, to ensure you don't overdrive whatever audio system or software Unify's output is routed into.

Over time, you may want to create your own Master Effects layer presets, tweaked to your liking.

Sharing effects chains using AUX Effects layers

Although you CAN put chains of audio-effect plug-ins on each individual Instrument layer, this isn't always ideal, and you are more likely to follow traditional mixing-desk practice and put effects chains onto separate AUX layers. Unify's AUX-effects layers are what would be called *Auxiliary buses* on traditional mixing desk (or in any DAW whose design is based on one).

- Add one or more (up to four) AUX layers to your patch using the **add-layer button** below the Unify logo.

- Add effects plug-in instances to your AUX layers in the usual way
- Drag to reorder the effects chain if necessary
- Use each effect's GUI to set its parameters to taste

*As soon as you add the **first** AUX layer* to your patch, every Instrument layer's **mix controls** section will immediately expand to include circular **send fader controls** like this:

- The fader marked “D” adjusts how much of the layer's output is fed *directly* to the Master Effects layer
- The other faders adjust how much of the layer's output goes to each of your AUX layers
- NOTE 1: the AUX layer's own *pan* and *level* fader settings determine how much that layer's processed output contributes to the final mix
- NOTE 2: you can Mute or Solo AUX layers, just the same as Instrument layers

Using MIDI layers

MIDI layers allow you to use *MIDI effects* such as arpeggiators, with a MUCH simpler work-flow than you'll find in most DAWs.

To use, say, the bundled BlueARP arpeggiator plug-in on a patch:

1. Add an empty MIDI layer, then add the BlueARP plug-in (OR use any of the pre-built BlueARP layer presets)
2. On each Instrument layer you want BlueARP to drive, use the **midi input selector** (below) to change the input setting from “IN” to your new MIDI layer.

That's all there is to it. Note the following about MIDI layers:

- You can *chain* MIDI effects together, so the output of one drives the input of the next
    - A typical case is putting an instance of the built-in MIDI Sustain effect before an instance

of BlueARP (which does not support MIDI CC#64–sustain pedal)
    - If your keyboard technique is not good, you might also want to use the built-in RipChord Player or some other chord-generating MIDI effect before an arpeggiator, to generate arpeggiated chord patterns.
- It can even be useful to use *empty* MIDI layers (no plug-ins at all, just the MIDI processing available on each layer), to set up, say, a key- or velocity-split to control multiple Instrument layers simultaneously.

The MIDI Sustain effect is useful when you have plug-ins which ignore MIDI CC#64 (sustain pedal), or don't respond to it as you might expect. This effect filters CC64 messages out of its input stream entirely, and delays sending note-OFF events while the pedal is DOWN.

Adding Drums and Percussion

The Unify Standard Library includes a small collection of **drum-kit** and **drum-loop** sample-sets for the built-in Guru Sampler instrument, found in the “Unify Bonus” collection.

- A *drum KIT* is a sample-set where each key on the keyboard (perhaps only *part of* the key-range) triggers a different “drum hit” sound.
- A *drum LOOP* is usually a single sample, which is a recording of a complete drum rhythm over one or more measures, at a specific BPM tempo.

A drum KIT can be triggered by an arpeggiator, to play rhythms at any desired tempo. A drum LOOP would not be used in this way, and can normally only be played back at its original tempo.

When playing drum loops in *Guru Sampler*, you would normally set the playback to **Unpitched** in the Master section of the *Guru Sampler* GUI. If you leave it at **Pitched**, the loop will play back at different speeds, depending on which key you strike, but will also change pitch, resulting in an unnatural sound. On rare occasions, this might be useful for creative effect.

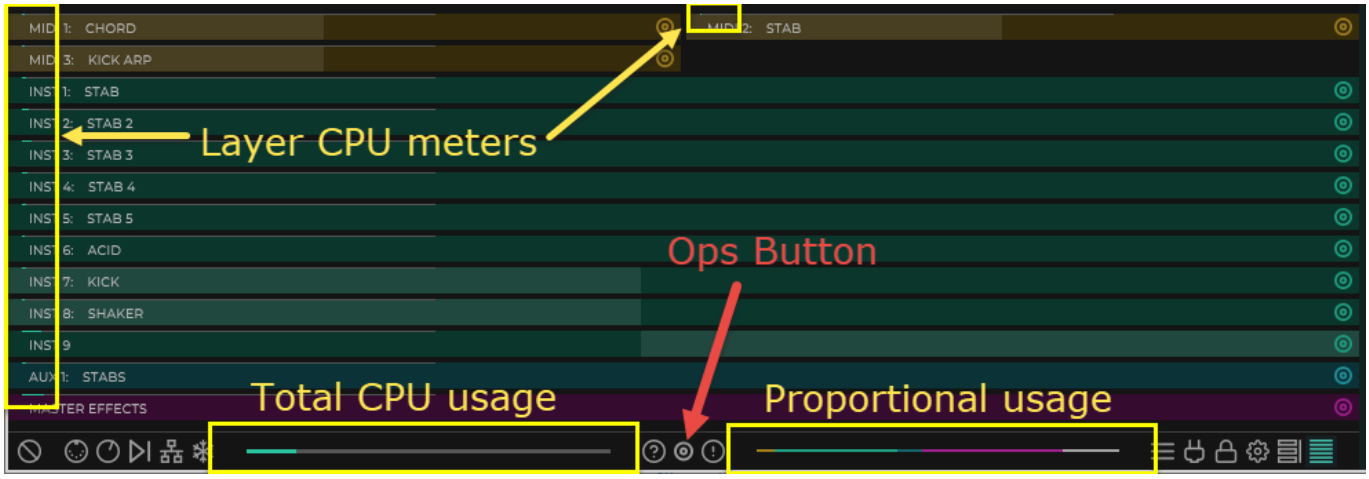
Understanding Unify's CPU meters

This page describes a new Unify feature which hasn't been released yet.

CPU meters have been added to Unify at v1.0.10. They are OFF by default; use the **main ops button** (center, bottom of the GUI) to toggle them on and off.

There are three kinds of CPU meters:

1. **Each layer has its own meter**, displayed in the top-left corner of its title bar
2. **Total CPU usage** is shown on the left side of the icon strip at the bottom of the GUI
3. **Proportional usage** is shown on the right side of the icon strip.



The **Layer** and **Total** meters show the percentage of available CPU time used in GREEN, and if this ever goes above 100%, a small part on the right will indicate the excess in RED.

The **Proportional usage** meter is always completely filled with color, with the lengths of the five color bars indicating how much of the total usage (green portion of the **Total CPU usage** meter) is accounted for by MIDI, Instrument, AUX, and Master layers, with a final off-white bar for “overhead” (everything else).

Showing and hiding the CPU meters

The CPU meters are hidden by default. Use the **main ops button** (two concentric circles at center, bottom of the GUI) to toggle the CPU meters on and off.

- A single click on the ops button pops up the “main operations menu”; “Toggle CPU meters” is the first item.
- As a shortcut, click the ops button while holding down the ALT button (Windows) or OPTION button (Macintosh)

What the Layer meters tell you

Computer audio systems process audio data in chunks called “buffers”, each containing a certain

number of samples (e.g. 256), which corresponds to a certain amount of time. For example, at 44100 samples/second, 256 samples represents about 5.8 milliseconds; we can call this the “buffer time”. To ensure no interruptions, each buffer MUST be processed in less than this amount of time. *How much less* it actually takes is a good measure of performance of the system.

Each of Unify's **layer** CPU meters indicate how long that layer takes to process, *as a percentage of the available buffer time*. As the percentage gets longer, the length of the green bar gets longer. At 100%, the green bar reaches almost all the way to the right-hand end, and a tiny bit of red will be shown. Above 100%, the amount of red will get progressively longer, maxing out at 150%.

In general, **short green bars are good, and red is bad**. An occasional flash of red might be OK, but meters consistently “in the red” indicates overload, which you will hear as crackling and drop-outs in the output audio.

What the Total CPU usage meter tells you

The **total CPU usage** meter tells you the fraction of buffer time Unify *as a whole* is using up. Due to Unify's use of *multi-threading*, this will not be quite as simple as the sum of all the layer fractions.

Unify processes each buffer in up to four stages as follows:

1. All MIDI layers (if any) are processed simultaneously, in separate threads.
2. All Instrument layers are processed simultaneously, in separate threads.
3. All AUX layers (if any) are processed simultaneously, in separate threads.
4. The *final mix* is computed, and the Master Effects layer is processed.

The amount of time required for each stage is, roughly, the amount of time taken by the *slowest layer* involved. Think of all the layers of each type as running a race: they all start at the same time, but the “winner” isn't important, because the next stage can't begin until they are all done, i.e., until the one in “last place” is finally done.

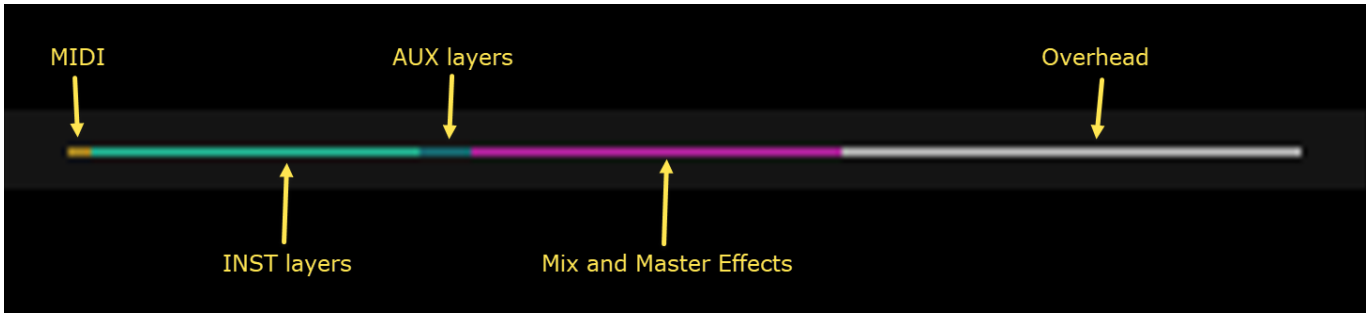
In addition to the measured per-layer times, there is also a certain amount of “overhead” for thread management and copying data into, and out of, each layer.

What the Proportional usage meter tells you

The **proportional usage** meter indicates how the total CPU usage fraction (shown by the **total CPU usage** meter) is divided, using colors similar to those used for the corresponding layers:

- BROWN represents the time to process all the MIDI layers

- LIGHT GREEN represents the Instrument layers (including any insert effects on each)
- DARK GREEN represents the AUX layers
- PURPLE represents the MASTER EFFECTS layer
- The OFF-WHITE portion at the right represents OVERHEAD



If the **total CPU usage** meter is showing heavy activity, the proportional meter and the individual layer meters can help you identify the cause.

When the **total CPU usage** meter shows only light activity (under 25%), you will tend to see a much larger percentage allocated to “overhead”, but this is basically meaningless.

What the CPU meters REALLY tell you

Measuring CPU usage in modern multi-core computers is *complicated*. Multiple cores can work at the same time, so multiple *threads* (units of work designated by the program as candidates for simultaneous execution) to run simultaneously. This sounds wonderful in theory—you think “my CPU has 8 cores, so I should be able to run 8 heavy-duty synth plug-ins at once—but in practice it's not that simple:

- A modern operating system has *thousands of threads* competing for the handful of CPU cores.
- Threads are assigned *priority levels*, so “high-priority” threads (like audio processing) generally take precedence over lower-priority threads (like flushing data from RAM to disk), BUT they can't do so forever—every thread's priority gradually rises as it gets denied CPU time again and again, until eventually its request is granted no matter what.
- Threads almost never run to completion, because even operations like reading or writing RAM are so slow, in comparison to operations within a CPU core, that it makes more sense for the core to switch to another thread while waiting for them to complete. (The cores only operate directly on data in the CPU's faster *cache memory*.)

When Unify's CPU meters indicate that certain processes are taking, say, 10% of buffer time, all this really says is that was the time it measured between *when the operation started* and *when it finished*. How much of that time was actually spent on audio processing, and how much was spent waiting for the CPU as it worked on other tasks, is impossible to know. In order of decreasing priority, these “other tasks” might include:

- Other audio threads, e.g., other tracks in a DAW, or perhaps other audio-processing programs such as screencasting or networking applications.
- Other hardware-related processing, e.g., handling interrupts from USB, disk, and network interfaces.
- Any other CPU-intensive programs, which set their thread priorities fairly high to provide good GUI responsiveness.

- Every one of the *thousands* of active threads on the system at any given moment.

You must also be aware that “buffer time” may not be constant.

- The present *stand-alone* Unify app actually does use constant-length buffers. (You set the length in Audio/MIDI Settings.)
- *Most DAWs* use a similar system of fixed-length buffers, *most of the time*.
    - Image-Line FL Studio is the most important exception. Unless you tell it to use *fixed-size buffers* for Unify, its buffer sizes vary constantly, and can be as short as *one or two samples*.
    - Logic Pro X uses the buffer length you specify (e.g. 256 samples), *except when processing automation*, in which case it drops to only 16 samples. It sometimes also switches to much longer buffers (1024 samples) for “idle” tracks (not currently playing or recording).

Unify's *overhead* processing depends only on the *number and type of layers in a patch*, NOT the buffer size. Hence, overhead tends to dominate if you (or your DAW) use extremely short buffers.

- If you set your buffer length short enough, Unify will get completely bogged down with overhead, and CPU usage will quickly rise to 100% or more.
- When a DAW switches to shorter buffer lengths automatically (e.g. while processing automation), you may see the “overhead” fraction (white part of the **proportional** CPU meter) become noticeably larger.
- This is why *FL Studio* users MUST specify “fixed-size buffers” for Unify. When FL Studio decides to use ridiculously short buffer sizes (which it does all the time, for no apparent reason), Unify won't be able to keep up (due to the very overhead that makes it so efficient with normal buffer sizes).

Using Unify's Macro Knobs for real-time control

Unify provides a set of eight *Macro Knobs*, available in the Footer part of the GUI by clicking the knob icon on the left-hand side of the icon strip:

Note there are two distinct kinds of *ops buttons* visible in the Macro Knobs view:

1. Just above and to the right of each macro knob the an ops button for that specific macro
    - Click to pop up a menu for editing the list of parameters linked to that macro, or change the associated MIDI CC number (see below)
2. At the extreme right-hand end of the Macro Knobs view is an ops button for the view itself
    - Click to pop up a menu for saving/loading the state of all macros, knobs, CCs, etc. as a preset.

Editing macro knob labels

You can double-click any of the macro-knob labels to change the default name “Macro1” etc. to whatever you wish. As with all in-place text editing in Unify, either confirm your changes by hitting the Enter/Return key on your keyboard, or click anywhere outside the edit box to cancel.

Macro parameters concept

The basic concept behind the macro knobs is that every Unify patch can be adjusted in real-time in up to eight separate ways, each of which is represented by a different macro knob. The word “macro” is used because the knob can be linked to a whole *list of actual parameters*, including *Unify's own parameters*, such as layer pan and level settings, and also *all plug-in parameters*.

Each macro knob is a visual representation of an underlying *abstract macro parameter* (or simply “macro”) whose value ranges from 0 to 1, (or if you prefer, 0 to 100%. The connection between each knob (on the GUI) and its associated macro is fixed and cannot be changed, but you CAN change:

1. Which external MIDI CC control can be used to adjust the macro while playing, and
2. Which *actual parameters* the macro is linked with, and how the macro value is mapped to the parameter value

The Unify plug-in exposes all eight macro parameters to the host DAW, so they can also be controlled through *host automation*.

Adjusting macro values: 3 different ways

Adjusting a macro value updates the values of all linked parameters. Macro values can be adjusted in three different ways, as follows:

- The GUI knobs can be adjusted by clicking and dragging with the mouse
- Each knob can optionally be linked to an associated MIDI CC number
    - See below for details
    - By default, the first four macro knobs are assigned to MIDI CC's 1, 2, 4, and 67, for compatibility with existing PlugInGuru patch libraries.
- In plug-in versions of Unify, macro parameters are exposed to the plug-in host (e.g. DAW)
    - See Host automation

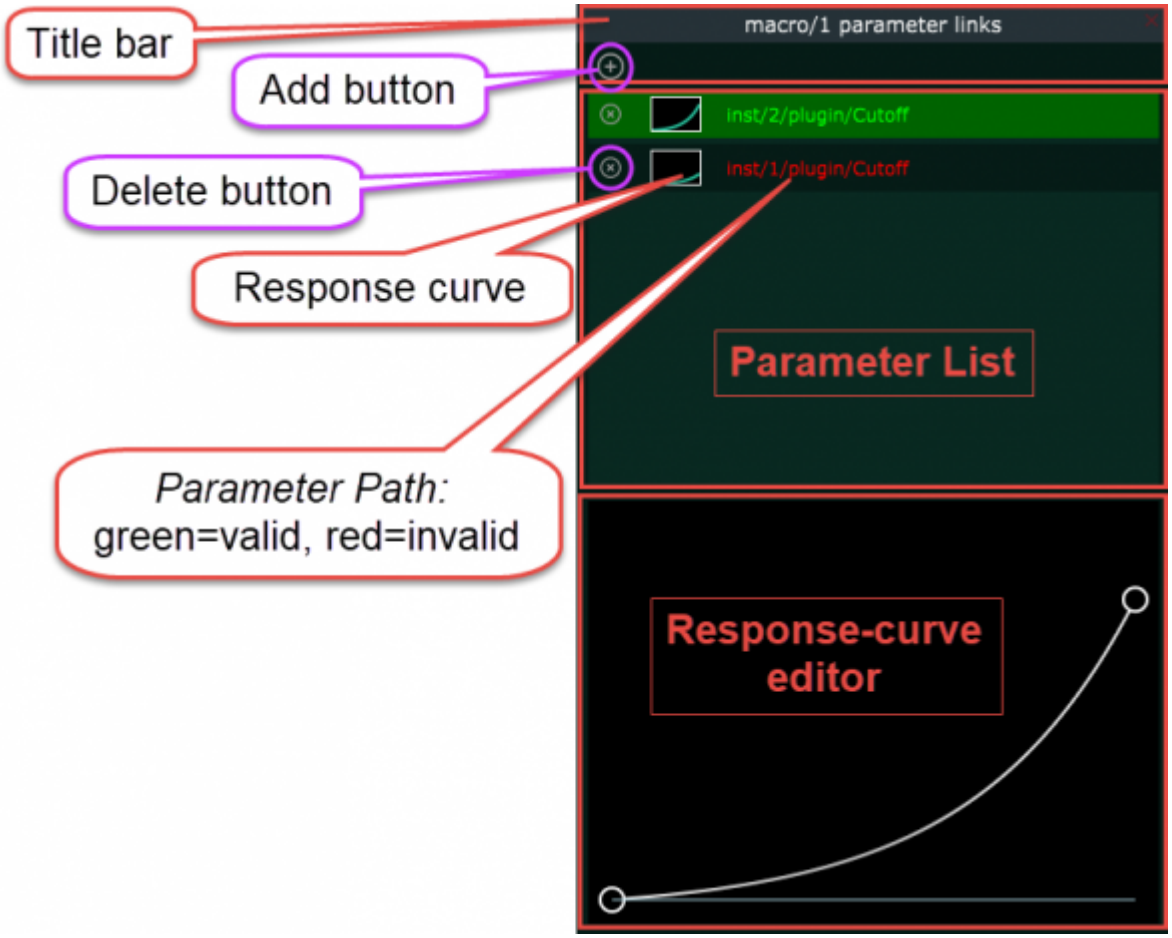
Editing the linked-parameters list for a macro

To edit the list of parameters linked to any macro knob:

- click the *ops button* just above and to the right of the knob, and choose “Linked Parameters…”, OR
- right-click the knob itself (same as clicking the ops button), and choose “Linked Parameters…”, OR
- Option-click (Alt-click on Windows) the knob to bypass the ops menu and open the links editor.

As shown in the screenshot below, the linked-parameter editor window is divided into four sections:

1. The **title bar** identifies which macro number the window is for (e.g., “macro/1” through “macro/8”).
    - Click the red X at the right-hand end of the title bar–note it's not very bright and may be hard to see on some monitors.
2. Just below the title bar is an area containing the **Add button**, for adding another linked parameter
3. The **parameter list** area has one horizontal strip for each linked parameter
    - At the left side is a **delete button**; click to remove that link from the list
    - Next is a small diagram representing the **response curve** for that parameter
    - The rest of the strip contains the parameter path
4. At the bottom is the **response-curve editor**, which works much like a velocity-curve_editorlayer velocity-curve editor.



Note that you can *resize* these windows (drag any edge) and also *re-position* them (drag the title bar) to suit your monitor layout. All windows will open in the same spot near the top-left corner of the screen, and with the same initial size, so you can easily end up with older ones hidden by newer ones if you don't move them.

The basic work-flow for editing a linked-parameters list is:

- **Add a new parameter to the list** by clicking the **add button** (plus sign in a circle), and choosing the desired parameter from a system of pop-up menus
- Click (*single-click*) any parameter in the **parameter list** to activate the **response-curve editor**, so you can modify how (input) values of the macro are mapped to (output) values of the parameter itself.

Understanding parameter paths

Unify uses character-strings called *paths* to select linked parameters in a hierarchical way. Each path is a sequence of *elements* separated by the "/" (forward slash) character. Elements may contain embedded spaces (e.g. *Osc1: Semitone*), but if a parameter name contains the "/" character, it must be *quoted* using *single quotes*, e.g. *ARP up/down* would have to be specified as '*ARP up/down*'.

Successive elements in the path narrow down the scope. For the first level, you can choose e.g. *master* (to select the Master FX layer), *inst* (to select an instrument layer), *midi* (MIDI layer), or *aux* (Aux FX layer). If you choose *inst*, the next element must be a layer number, e.g. 1, 2, etc. Subsequent elements further refine the choice. For example:

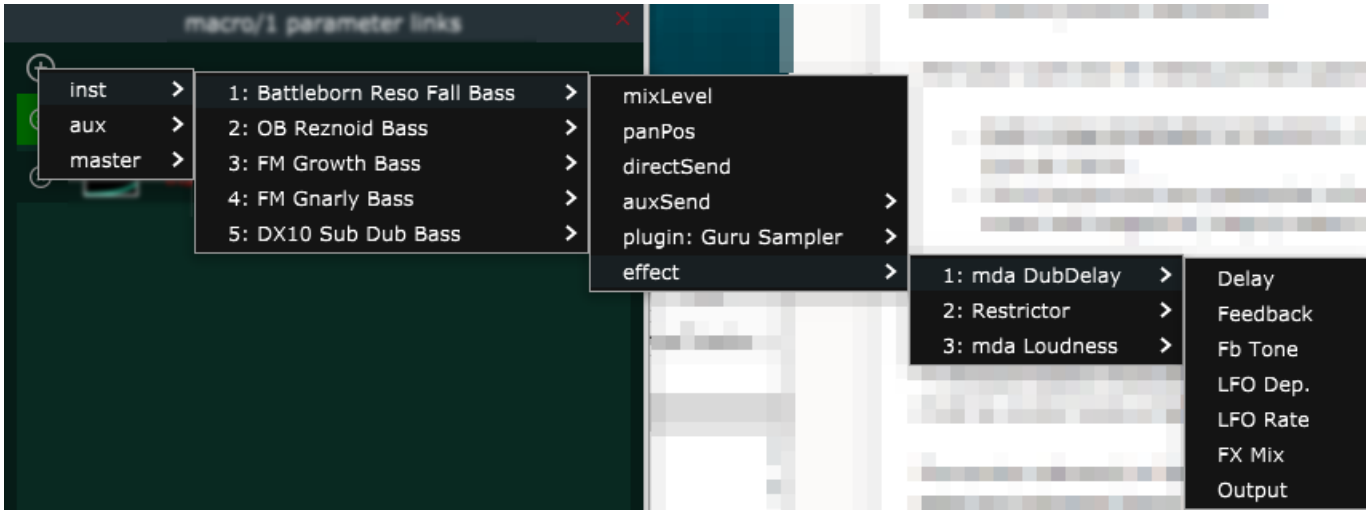
- *inst/1/panPos* refers to the the pan setting of the first instrument layer; this is one of Unify's own parameters.
- *inst/1/plugin/Cutoff* refers to the *Cutoff* parameter of the instrument plug-in for Unify's first instrument layer; this is a *plug-in parameter*, not a Unify parameter.

You don't need to worry too much about the details of parameter paths, because the “plus” icon at the top of the link editor window brings up a cascade of menus which allow you to select only valid paths.

Because parameter paths are just character-strings, they may become *invalid* if you change the Unify patch, e.g., by deleting layers, swapping or removing plug-ins, etc. Invalid parameter paths display in RED in the link editors, but they are otherwise harmless, i.e., Unify simply ignores them. You would normally wait until you have stabilized your patch structure before starting to define macro parameter links.

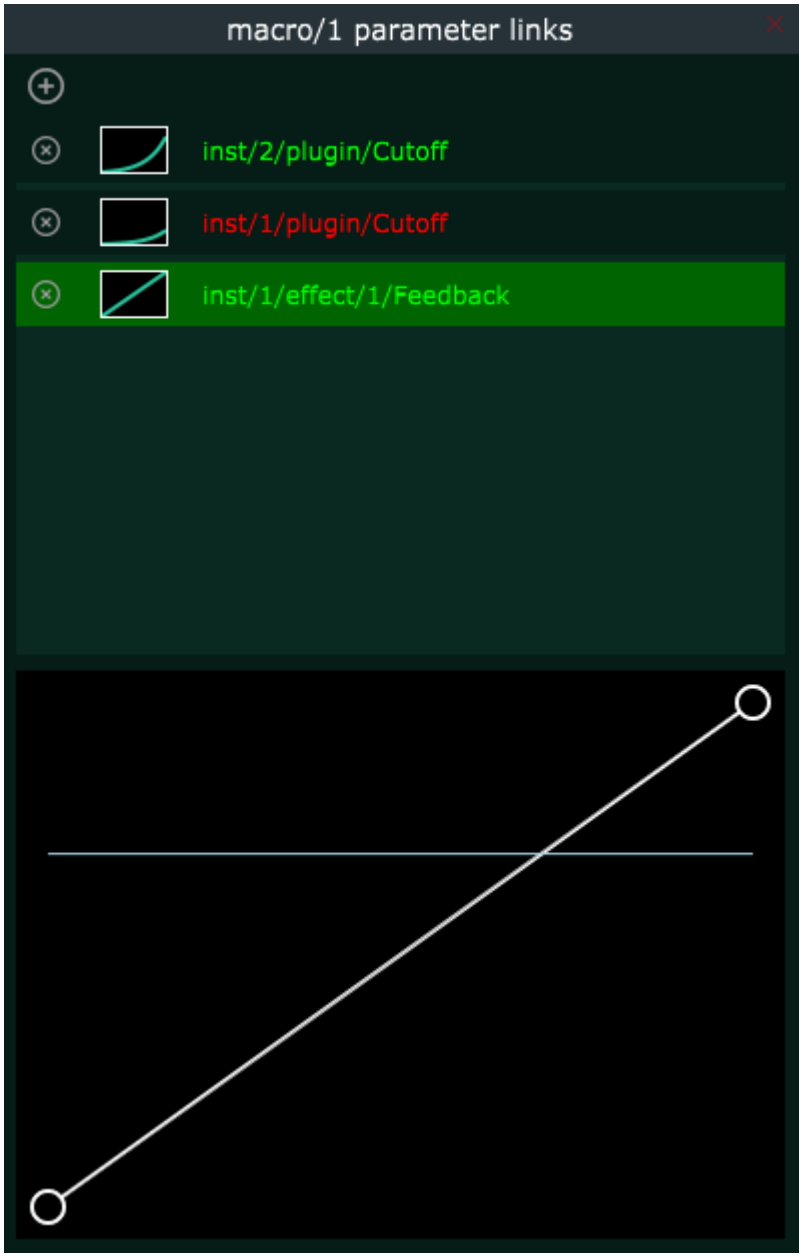
Adding a parameter

To add a linked parameter, click the **add button** (circled plus-sign icon) just under the title bar, to pop up the first in a hierarchical series of menus.



- The first menu level has one entry for each type of layer in the patch, e.g. *midi*, *inst*, *aux*, *master*. Note the *midi* and *aux* items won't appear at all if the current patch has no MIDI or AUX layers.
- After clicking, say, *inst* and moving the mouse pointer to the right, the second menu level will appear, with one entry per layer of the selected type. If you have entered your own titles for the layers, the layer titles will be shown as well, to help you navigate. If not, you'll just see the layer numbers.
- In the screenshot above, the mouse was positioned over the first layer (“1: Battleborn Reso Fall Bass”) and moved to the right, revealing the third-level menu, offering the following choices:
    - *mixLevel* and *panPos* settings for the layer
    - *directSend* and *auxSend* (a sub-menu with choices 1, 2, 3, and 4) settings for the layer
    - *plugin: Guru Sampler*, a sub-menu listing all the parameters which the layer's instrument plug-in (in this case, *Guru Sampler*) exposes for automation
    - *effect*, a sub-menu
- In the screenshot above, the *effect* item was followed, revealing the fourth-level menu, with one item for each audio-effect plug-in used on the layer

- In the screenshot above, the *1: mda DubDelay* effect item was followed, revealing a final sub-menu, listing all seven parameters which this particular plug-in exposes for automation.
- If you were to click, say, the *Feedback* item on this final menu, all menus would close and a new entry would be added to the parameter list with the path *inst/1/effect/1/Feedback*. Furthermore, that parameter would automatically be *selected* (highlighted in green), and its associated response curve would automatically be loaded into the **response-curve editor** at the bottom of the window.



Adjusting response curves

Macro parameter values are *fractions* in the range 0.0 to 1.0 (i.e. 0 through 100%). When a macro parameter is linked to an actual parameter with a different range (e.g. a mix level with a range of -60 to 0 dB), the linked parameter will be set to the corresponding *proportion* of its range.

In parameter response curves, the horizontal axis represents the value of the macro, with 0 at the left and 1 (or 100%) at the right. The vertical axis represents the corresponding value to be assigned to

the linked parameter *as a proportion of that parameter's total range*, with the minimum at the bottom and the maximum value at the top.

Click (*single-click*) any row in the list to *select* that parameter, so you can edit its response curve in the graph editor.

- A light blue horizontal line will be shown on the graph, indicating the **current parameter value**.
    - This is useful if you have already set the parameter to where you want it to start (or end).
    - You can then drag the start (end) point of the response curve up (down) to coincide with the line.
- Drag either endpoint (circles) up or down to change the output range
    - You can drag the right-hand endpoint *below* the left one for an inverted response (increasing the macro parameter value *decreases* the value of the linked parameter)
- Drag up/down on the line between the endpoints to adjust the shape of the curve (convex or concave)
- *Double-click* between endpoints to add a new “split point”
    - This splits the curve segment in two
    - Split points can be dragged up/down and left/right
    - Individual curve segments can be dragged up/down to adjust shape
- Double-click any split point to delete it
    - This merges two segments back into one

Editing parameter paths as text

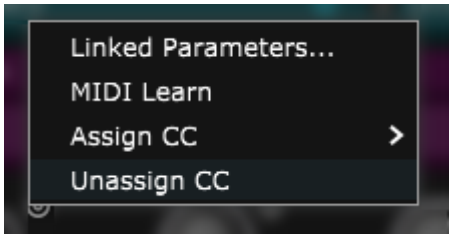
Once you get really comfortable with the parameter paths, you can also *double-click* a row in the list to edit the linked parameter's path as text.

- Use the mouse and keyboard as you would in any standard text editor
- Press the Enter/Return key on the keyboard to confirm the change and finish editing, OR
- Click anywhere outside the editor box to stop editing and revert to the original path

Linking MIDI CCs to macro knobs

Macro knobs can optionally be linked to MIDI CC's for real-time control. The linked CC number will be shown inside the knob like this:

To change a knob's CC assignment, click the knob's *ops button* (or right-click OR Ctrl+click the knob itself) to pop up the knob menu:



The “Linked Parameters...” item opens the linked-parameters editor as already discussed above. The other three items relate to CC assignment, and are listed below in reverse order (bottom to top):

- Select **Unassign CC** to remove any existing CC assignment.
- Use the **Assign CC** to select the new CC number from a list (sub-menu), OR
- Select **MIDI Learn**
    - The knob will show a question mark, indicating that Unify is waiting for a MIDI CC event
    - Adjust any CC knob or slider on your MIDI controller
    - When Unify receives the first CC event, the knob's CC assignment will immediately update, and Unify will drop out of “MIDI Learn mode”
    - If you select **MIDI Learn** by accident, or change your mind, just click the knob ops button again and select **CANCEL MIDI Learn** from the ops menu.

Macro-Definitions and CC-Assignments presets

At the right-hand end of the Macro Controls area you will find an “ops button”—a small icon of two concentric circles. Clicking on this opens a small menu for saving/recalling presets like this:

The menu options allow you to save the current *Macro Definitions* (which parameters are assigned to each macro, including response curves), or the current *CC Assignments* (which MIDI CC's are assigned to control each knob) separately.

Macro Definitions presets are useful if you set up a particular real-time control arrangement that you might want to use with multiple patches you're creating. When you create the next patch in your series, you can re-load the Macro Definitions preset.

CC Assignments presets allow you to quickly save and recall specific CC-to-knob linkages, which is particularly useful if you have multiple MIDI controllers. You could then switch between, say, an M-Audio keyboard whose knobs send MIDI CC's 16, 17, 18, and 19, and a Native Instruments Komplete keyboard whose knobs send CC's 14, 15, 16, etc.

Controlling Unify using DAW automation

NOTE This page is only relevant for *plug-in versions* of Unify.

Nearly all 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*, its *unit of measure*, etc.

Unify exposes its eight macro parameters for automation

The Unify plug-in exposes exactly eight automatable parameters, which correspond to the eight macro knobs. The host only knows that there are eight parameters, named “Macro1” through “Macro8”, 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 there is to tell about Unify and automation, other than a few minor details:

1. Automation changes to unused Unify macros (i.e., macros to which no actual parameters are linked) are ignored.
2. Even if you edit the macro-knob names in Unify, your DAW will still only see “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.

Potential problems using both MIDI CC assignments and Automation

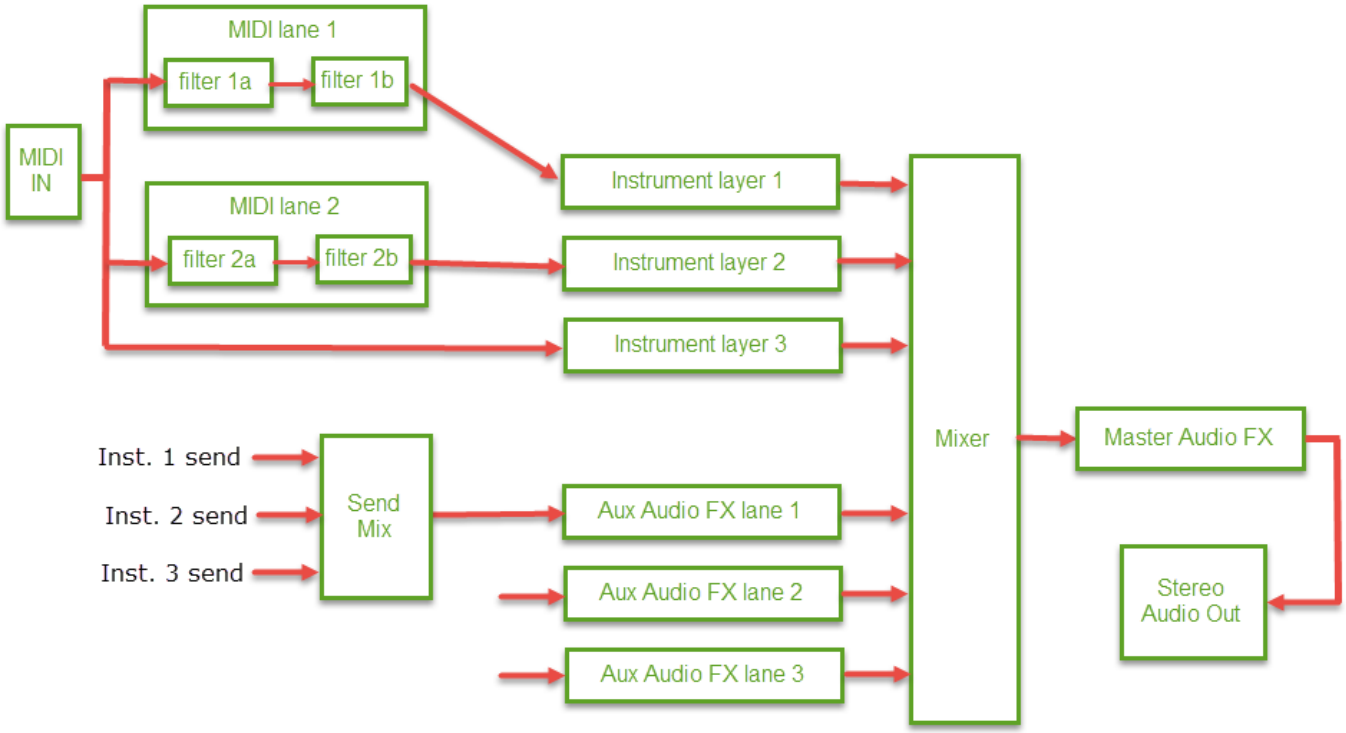
DAWs can record MIDI CC activity as well as automation. If you link any of your macro knobs to MIDI CCs, and manipulate those MIDI CC controls *while recording automation*, your DAW will record *both* the MIDI CC activity and the automation changes, and will attempt to send both into Unify on playback.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

We suggest that you use only MIDI CC recording in your DAW for all changes related to real-time musical expression (and be sure NOT to record automation at the same time), and use automation only for programmed changes such as fade-ins, fade-outs and other transitions.

How Unify processes MIDI data

Basic MIDI routing

This diagram (from Getting started with Unify: Overview) shows the basic signal routing in Unify:



For patches with no MIDI layers, incoming MIDI data (from your DAW and/or MIDI keyboard/controller) is always routed to ALL Instrument layers.

When there are MIDI layers, incoming MIDI data is always routed to ALL MIDI layers, and each Instrument layer can be set to receive MIDI either directly from the input stream, OR from the output of any selected MIDI layer.

In Unify v1.0.x, MIDI data is not routed to audio-effects layers (either AUX or MASTER). This may change in future.

MIDI routing in MIDI layers

As the above diagram shows, MIDI data is passed in daisy-chain fashion along a chain of MIDI effects. The MIDI input stream is the MIDI input for the first MIDI effect, the output of the first effect is the input to the next, and so on, and the output of the last MIDI effect in the chain is the output of the entire MIDI layer.

This chaining can be useful when you need to modify the input MIDI stream before passing it on to the next effect, e.g. using an instance of MIDI Filter to interpret MIDI sustain-pedal (CC#64) events ahead of an instance of BlueARP, which doesn't understand those events itself.

MIDI routing in Instrument layers

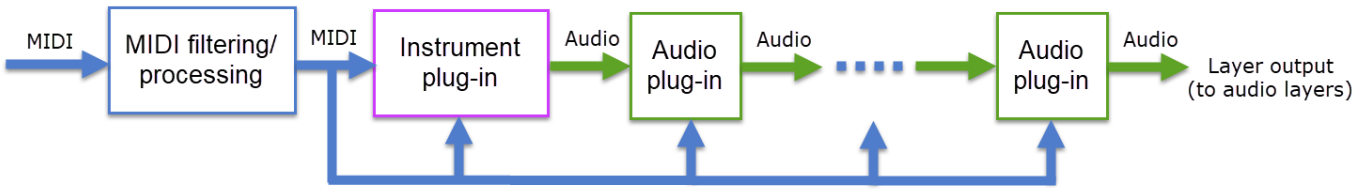
MIDI routing in Instrument layers is completely different from that in MIDI layers: The input MIDI stream is routed directly to EVERY plug-in in the layer, like this, and the layer does not have any MIDI output:

What this means is that not only the Instrument plug-in, but ALSO any insert effects on the layer, can be controlled via MIDI. This is particularly useful when using MIDI data-streams generated by MIDI generator plug-ins such as arpeggiators. Of course, only plug-ins that actually support MIDI control can take advantage of this feature. Nearly all instrument plug-ins support MIDI control (if only for playing notes), but quite a few audio effects do too. (Effect plug-ins by Polyverse/Infected Mushroom have particularly good MIDI control abilities.

MIDI processing built into MIDI and Instrument layers

The above descriptions of MIDI routing in MIDI and Instrument layers don't tell the whole story. Both MIDI and Instrument layers have quite a bit of built-in MIDI-stream processing.

The following diagram shows both MIDI and audio routing in Unify's Instrument layers:

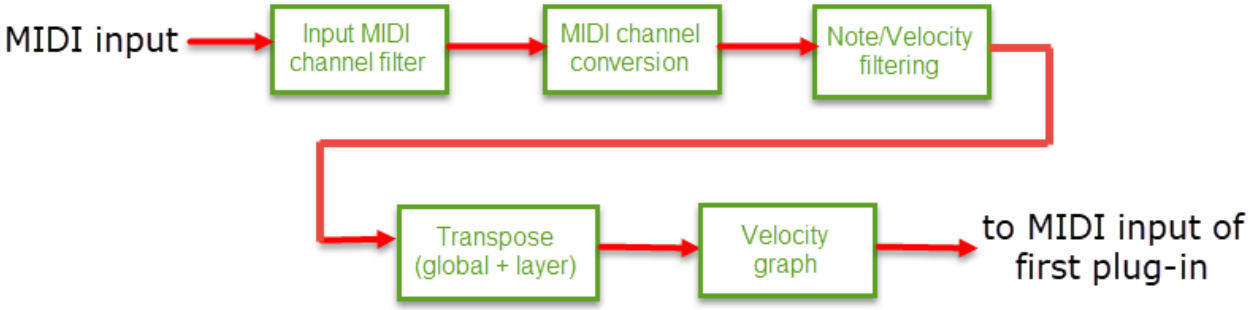


The box labeled *MIDI filtering/processing* on the left is represented in the Unify GUI by the “MIDI controls cluster” at the left-hand side of the layer, which consists of eight primary control rectangles plus two more (the *MIDI activity light* and *MIDI latch control*) which appear only when they are relevant.

The *MIDI Source* box (top left corner) determines where the layer's MIDI data comes from. This routing choice happens before the MIDI data even enters the *MIDI filtering/processing* box.

- **IN** selects the main input
- **M1** selects the output of MIDI layer 1
- **M2** selects the output of MIDI layer 2, etc.

The “MIDI filtering/processing” box in the diagram above is itself a sequence of five processing stages:



Stage 1: MIDI channel filtering

At the first stage, the *MIDI channel number* (1–16) of each incoming MIDI event is compared against the setting of the *Input Channel control*, and the event is only passed on to the next stage if it matches:

- **ALL** allows all events to pass (no MIDI channel filtering)
- **1** through **16** filter out events whose MIDI channel (1–16) does not match

Stage 2: MIDI channel conversion

The second stage allows the option to change the event's MIDI channel number. This could be useful if you have multiple MIDI keyboards assigned to, say, MIDI channels 4, 5, and 6, and you have set up MIDI channel-based filtering to route these to different layers, but you would like to change the channel numbers back to 1 before processing them, either because you're working with a plug-in which *only* works on MIDI channel 1, or you simply prefer not to set your individual plug-ins to specific MIDI channels. (You might have loaded the layer from a layer preset, which you have defined with all plug-ins set on MIDI channel 1 for simplicity.) Stage 2 is controlled by the *MIDI output channel* control:

- **NC** means “no change”, and is the setting you would most commonly use

- **1** through **16** cause all incoming MIDI events to be reassigned to that channel number

Stage 3: MIDI pitch- and velocity-based filtering

The third filtering stage is the basis for key- and velocity-splitting. By setting multiple layers to respond to non-overlapping *pitch* ranges, you can define *key-splits*. By setting two or more layers to respond to the same pitch range, but non-overlapping ranges of the MIDI note-velocity (1–127), you can define a *velocity split*, where the layer which plays depends on how hard you strike notes on the keyboard. Stage 3 is controlled by the four control boxes at the right-hand side of the MIDI-controls cluster:

The control box marked **G8** in the image indicates the upper limit of pitch; the box below it marked **C-2** is the lower limit. You can adjust these values in three ways:

- Click and drag up/down with the mouse
- Right-click (or Ctrl+clicking) to pop up a small edit box, and either:
    - Type a note name (e.g. “C3”) or a MIDI note-number (e.g. 60) and hit Enter, OR
    - Press the appropriate key on your MIDI keyboard, and the displayed note-name will change to match

The two control boxes on the right are similar, but define the upper (top-right) and lower (bottom-right) *velocity* limits. These can only be adjusted using the mouse (either click/drag or use mouse-wheel).

Stage 4: Pitch transpose

MIDI pitch values for key-splits are defined *prior to transposing*, so you can specify splits based on *physical* keys on your keyboard. (Technically, the key-split ranges are defined based on the *MIDI note-numbers output by your keyboard*, which might themselves be transposed if you have activated any transpose capability built into the keyboard controller itself.)

The fourth processing stage allows you to transpose the pitch at which the layer sounds by adding an offset (in semitones, which may be negative) to the note-numbers for each MIDI note-on or note-off event. The offset is the sum of the **layer transpose offset** and the **global transpose offset**.

The **layer transpose offset** affects only the given layer, and is is set by the *layer transpose control* at the bottom-left corner of the MIDI-controls cluster:

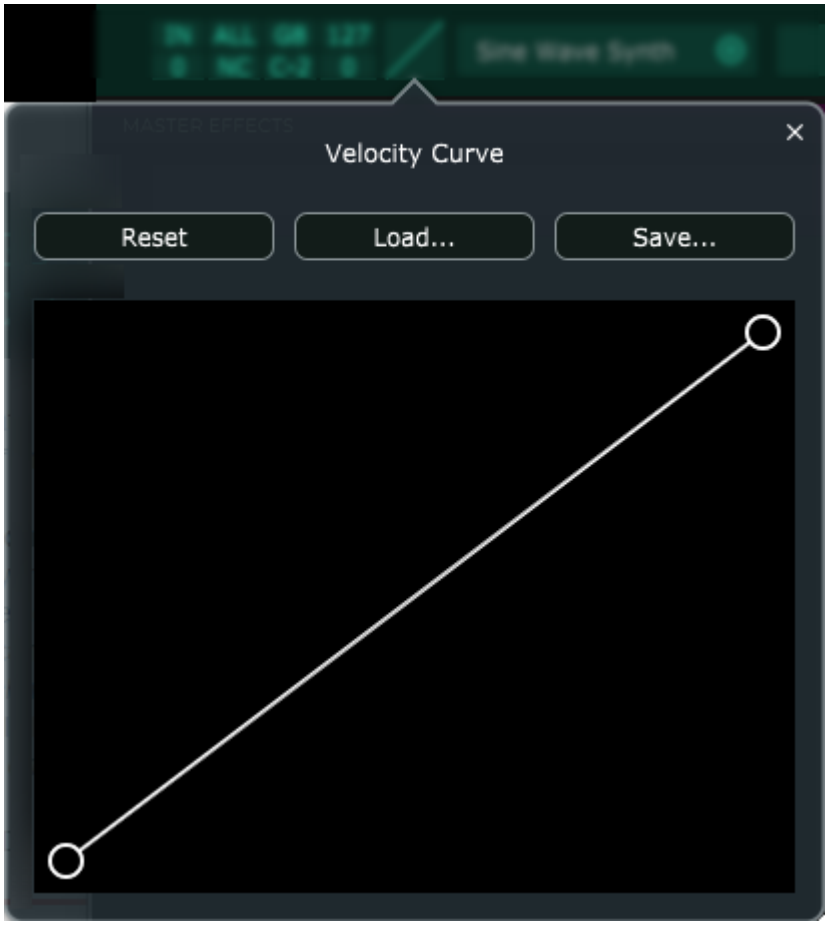
The **global transpose offset** affects ALL instrument and MIDI layers, and hence allows you to transpose the entire composite Unify “instrument” up and down. The global transpose offset is set by the *global transpose box* at the bottom-right corner of the Show MIDI view which appears in the Footer of the Unify GUI window when you click on the MIDI icon in the Icon Strip below:

Both the global and layer-transpose boxes can be adjusted in the same three ways:

- Click and drag up/down with the mouse
    - If you hold down the Alt key (on a PC) or the Option key (on a Mac), the value will change by octaves
- Right-click (or Ctrl+clicking) to pop up a small edit box, enter a value in semitones (may be negative) and hit the Enter key.

Stage 5: Note-velocity modification

Finally, the fifth stage of layer MIDI processing allows you to apply an arbitrary *response curve* to the velocity values associated with each note-on event. Clicking on the **MIDI Velocity Graph control** at the right-hand side of the MIDI-controls cluster pops up a *curve editor window* like this:



The graph itself represents *incoming* MIDI note-velocity values along the horizontal axis, with lowest (quietest) velocities on the left and highest (loudest) velocities on the right. The vertical axis represents *outgoing* (processed) velocity values, lowest at the bottom and highest at the top. The default velocity curve is a straight line–the so-called “identity mapping” where each incoming velocity value is mapped to the identical value (no change at all).

- The **Reset button** reverts the curve shape back to the straight-line “identity” curve, as shown
- The **Load… button** lets you load a previously-saved curve-shape preset (XML file)
- The **Save… button** lets you save the current shape as a new curve-shape preset

In the graph control itself, you can:

- Click and drag the *endpoints* (circles) up and down
- Click anywhere *between* endpoints and drag up/down to change the curvature of that segment of the curve
- Double-click anywhere *between* endpoints to *create a new split point*, splitting the segment into two
- Single-click directly *on a split point* and drag to move it up, down, left, or right.
- Double-click directly *on a split point* to *delete it*, joining two segments back into one

For velocity curves, you will rarely need to create split points. You will usually only need to adjust the curvature (*slightly* – a little change goes a long way). Dragging upward to create a *convex* curve will make the layer *more responsive to velocity*, and can be useful when you are working with a weighted-key MIDI controller. Dragging downward to create a *concave* curve will make the layer *less responsive to velocity*, and can be useful when playing a very lightweight synth-action keyboard.

Show MIDI view

The *Show MIDI view*, which appears in the Footer part of the Unify GUI window, allows you to visualize MIDI events coming into Unify from a MIDI keyboard (or played back from a MIDI track in a DAW). You can also *generate* MIDI events by clicking on some of the controls in this view, which can sometimes be useful for testing, but note that generated MIDI events are internal to Unify and **cannot be recorded in your DAW**.

To see the *Show MIDI view*, click on the *MIDI connector icon* at the left end of the Icon Strip.

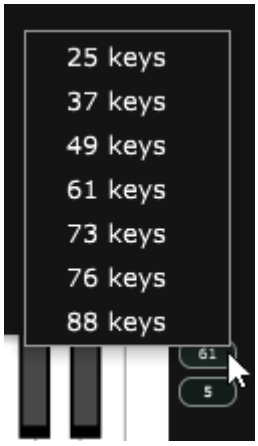
The Show MIDI view consists of a group of vertical-bar controls at the left, a keyboard, and three small controls at the right-hand side.

Keyboard

The *keyboard* area allows you to visualize MIDI note-on and note-off events, as the keys will light up in blue when notes are played. Just below the keyboard is a long, thin horizontal strip which lights up bright pink when the MIDI sustain-pedal (CC#64) is depressed. While the sustain-pedal is down, active notes on the keyboard will change from blue to pink, indicating that those notes are being held (sustained).

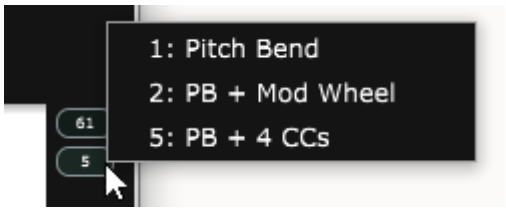
You can also click notes on the keyboard with your mouse to generate MIDI note events, but note these are internal to Unify and **cannot be recorded in your DAW**.

The *width* and *range* of the keyboard can be set by clicking on the top button at the very right-hand end of the Show MIDI view, which pops up a small menu like this:

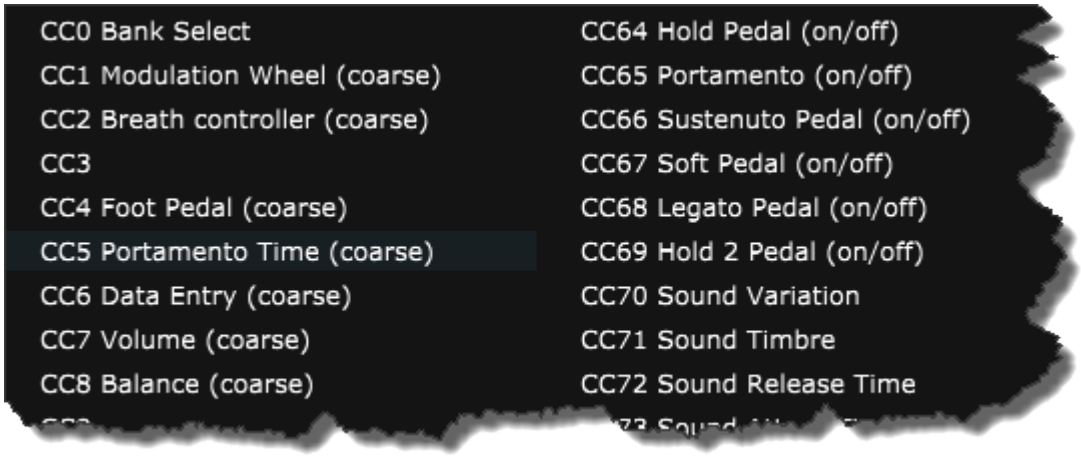


Pitch-bend and CC strips

To the left of the keyboard, you will see a number of vertical strip controls, which indicate MIDI *pitch-bend* and *continuous-controller* (CC) activity. How many strips appear depends on the setting of the button in the middle at the right-hand end of the Show MIDI view, which pops up a small menu when clicked:



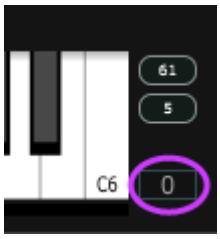
The leftmost strip indicates pitch-bend activity. The others are for specific MIDI CC's, and each one has a number at the top to indicate which MIDI CC it is monitoring. Right-clicking (or Ctrl+clicking) anywhere in the CC strip pops up a large menu from which you can change its assigned CC number:



You can also (left-)click and drag up/down in any of the vertical strip controls to *generate* corresponding MIDI pitch-bend and CC events. These are internal to Unify and **cannot be recorded in your DAW**.

Global Transpose control

At the very bottom-right corner of the Show MIDI view you will find the *Global Transpose control*, which allows you to define a signed offset (in semitones) to be applied to ALL Unify MIDI and Instrument layers, and thus transpose the entire instrument.

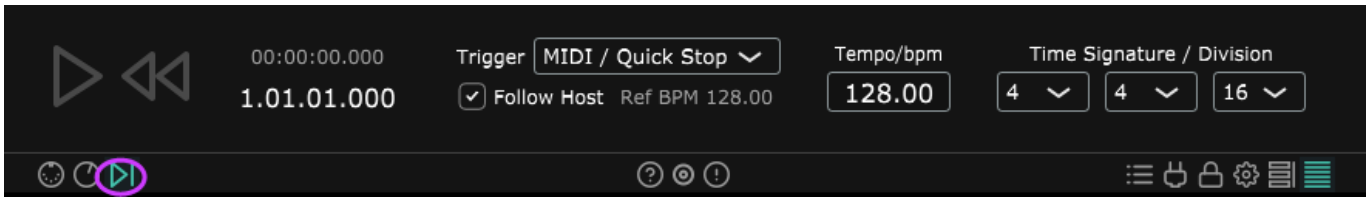


The Global Transpose control can be adjusted in the same three ways as the transpose boxes found in

Using the Unify Transport

Unify is a *plug-in host*, much like a DAW, and thus it has its own internal timing/clock mechanism, called a *Transport* (by analogy to a mechanical tape-transport). Unify's transport provides timing information to any plug-ins that may require it (primarily MIDI arpeggiator effects and synthesizers with arpeggiator functions).

To see the Transport in the Footer area of the Unify GUI, click on the *transport icon* (looks like a triangle with a vertical line beside it) at the bottom left part of the Icon Strip:

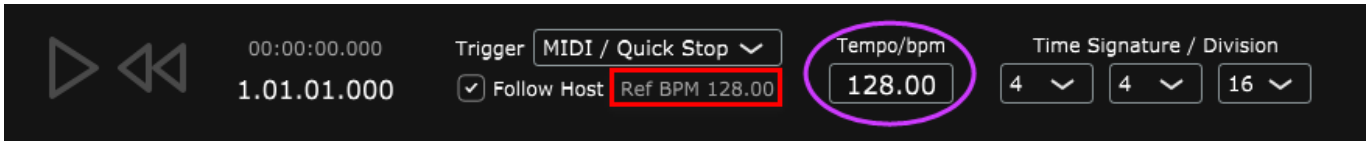


Tempo, Time Signature, Song Position

Unify's Transport knows about three distinct, but related pieces of timing information:

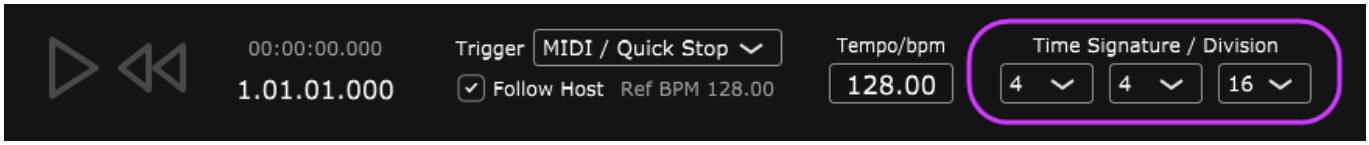
1. The current **tempo** (aka BPM or beats-per-minute rate)
2. The **time signature**
3. The current **song position**, which is the current location along a musical time-line (e.g. DAW track)

Tempo is pretty easy to understand—it measures how quickly beats go by–and is obviously critical to software functions like arpeggiators and tempo-synced delays, to ensure that they play at the same speed. The Transport's current tempo value in beats per minute (BPM) is displayed in the *tempo/bpm indicator*, here outlined in purple:

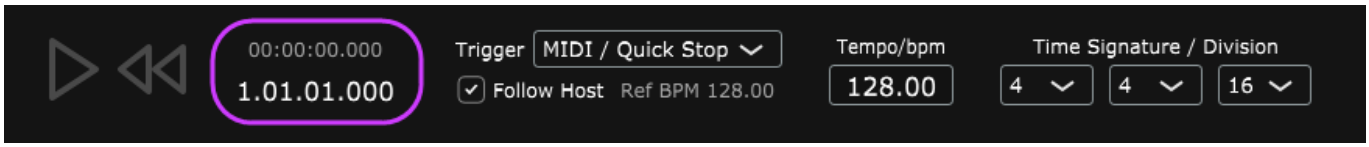


So-called “BPM” patches in Unify have an associated *Reference Tempo*, which is the rate at which the sound designer intended them to be played. This is displayed as indicated by the red outline above. (Non-BPM patches will display “No ref BPM”.)

Time signature is a concept from music theory, and simply describes how beats are grouped into *bars* (measures). A time signature consists of a *numerator* which specifies the number of beats per bar, and a *denominator* which specifies the kind of note or rest that represents each beat. A song in “3/4 time” has three beats per bar (numerator=3) and each beat is a quarter-note (denominator=4). The Transport view in Unify provides pop-up menus for numerator and denominator, plus a third one for “division”, which will be explained in a moment.

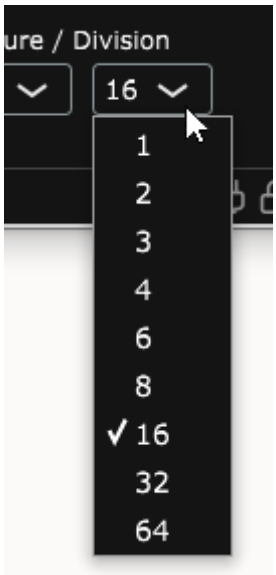


To keep a group of arpeggiators playing together, tempo and time signature are not sufficient. They must all be at the same bar/beat position, which is referred to as **Song Position**. The Unify Transport view displays the current song position in a pair of numeric readouts as shown:



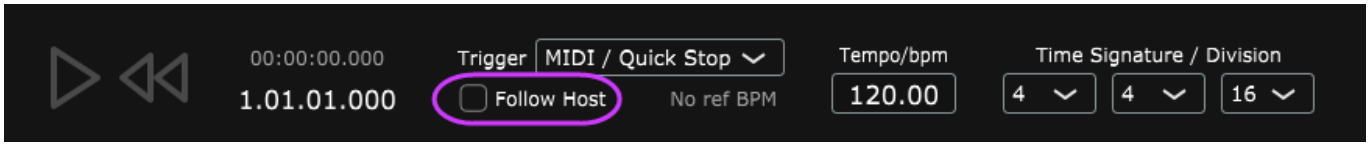
The smaller readout at the top shows the song position in hours, minutes, and seconds (with three digits of fraction after the decimal point, i.e., milliseconds). The larger readout below indicates **bar number**, **beat number** (within the current bar), and **beat division**, which is an arbitrarily-chosen subdivision of the beat, and again there are three digits of fraction following the decimal point.

The beat division is 16 (i.e., 16 divisions per beat) by default, but you can use the rightmost pop-up menu to change it to any power-of-two value from 1 to 64:



Follow Host

When Unify is running as a plug-in inside a DAW, you can check the **Follow Host** checkbox to synchronize it to the DAW's own Transport.



When **Follow Host** is checked, the **Tempo** and **Song Position** indicator will automatically update when the corresponding settings are changed in the host DAW. (Note that, for BPM presets, the displayed “Reference BPM” number will remain unchanged even when the DAW's tempo changes.)

Sometimes, you may want to de-couple Unify's transport from the host's, so that you can e.g. audition BPM presets at their reference tempo. To do this, simply un-check the **Follow Host** checkbox.

NOTE the **Follow Host** checkbox is displayed in the stand-alone Unify app, but this is only to allow

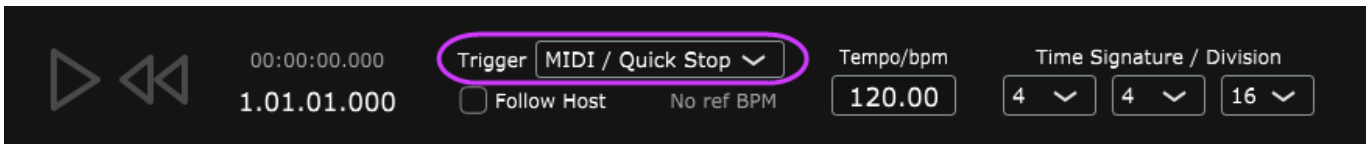
you to specify whether **Follow Host** should be checked or not, when creating and saving BPM patches.

Starting and stopping the Transport

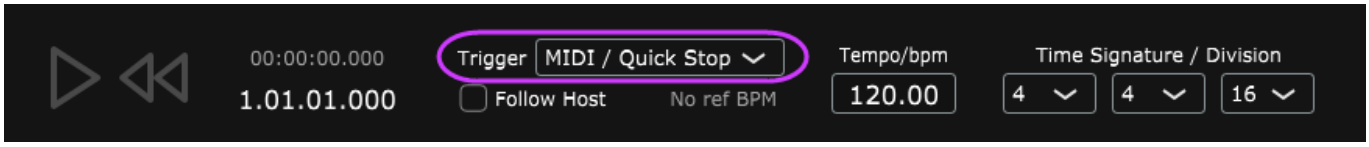
Unify's transport can be started (and stopped) in three very different ways.

Method 1: In plug-in versions of Unify, when **Follow Host** is checked, the Transport will automatically start and stop when the host's Transport does.

Method 2: When the **Trigger** menu is set to “MIDI / Quick Stop” (the default), the Transport will start automatically as soon as a MIDI note is played, and stop immediately when NO notes are active (i.e., when a matching MIDI note-off event has been received for all previous note-on events).



Method 3: When the **Trigger** menu is set to “Manual start/stop”, the two large icon buttons at the left-hand side of the Transport will light up to indicate they are active (see image, red outline).

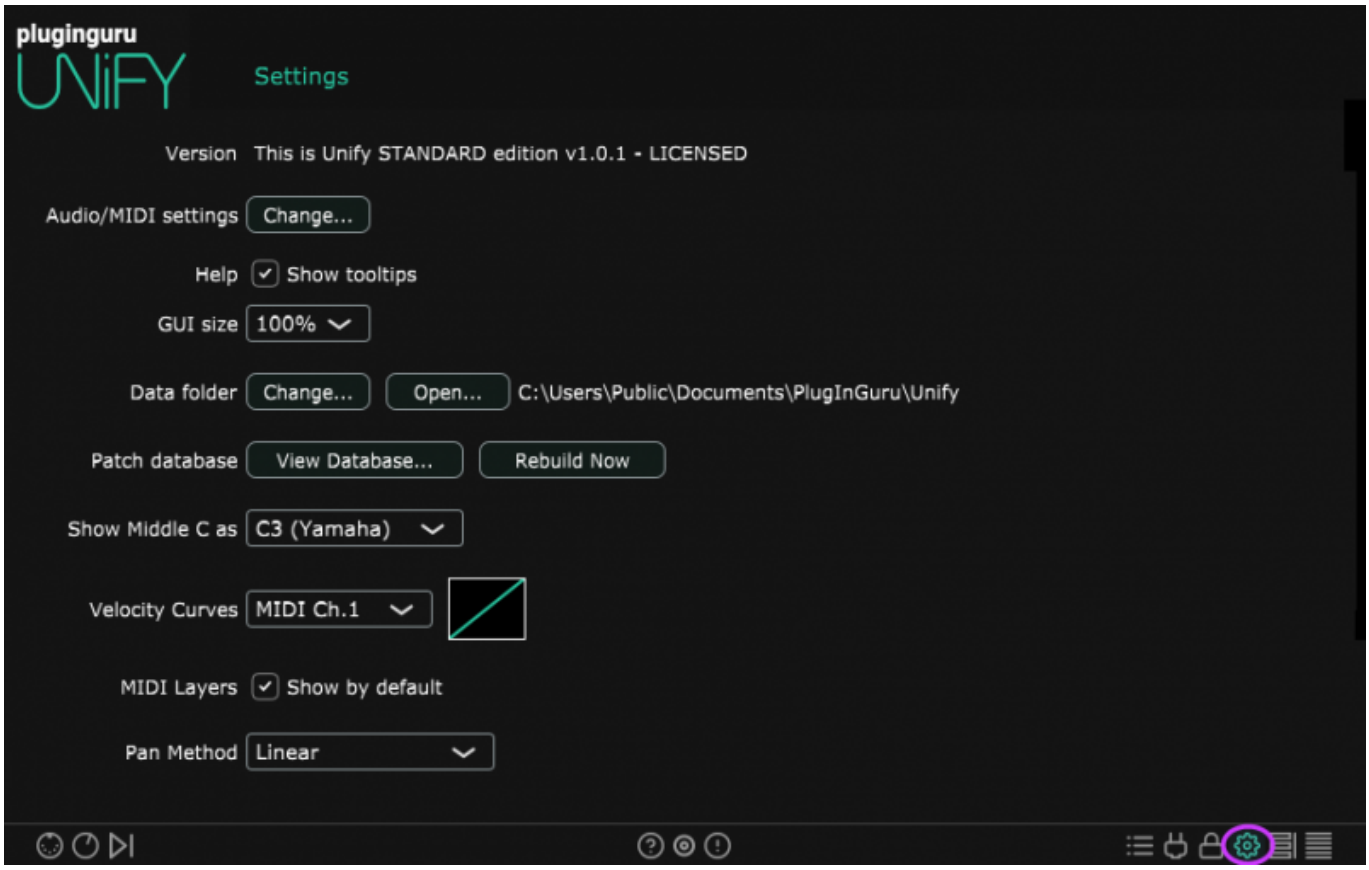


- Click on the double-triangle “Rewind” icon to rewind the Transport (reset the Song Position)
- Click on the triangular “Start” icon to start the Transport
    - It will change to a double-bar “Pause” icon: Click on this to pause the Transport
    - The double-triangle “Rewind” icon will change to a square “Stop” icon: Click this to stop AND rewind

Using the Settings view

Unify's Settings view allows you to change several important aspects of the program's operation, which are saved so they affect all subsequent instances of both plug-in and stand-alone versions of Unify.

To open the Settings view, click on the Settings icon at the bottom right—it looks like a gear.



From top to bottom, the individual items are:

- **Audio/MIDI settings**
  - Click “Change…” button to choose MIDI and audio settings for the stand-alone app
- **Show tooltips** checkbox
  - *Tooltips* are little floating text windows which appear when you hover the mouse pointer over various controls, explaining briefly what the control does.
  - These can be very helpful while you're still learning the Unify GUI, but at a certain point you'll want to turn them off.
- **GUI Size** menu
  - Allows you to select a *magnification factor* for the GUI
  - Higher magnifications yield larger, crisper fonts and graphics—very nice on high-density displays, and a boon for folks with less than 20/20 vision
  - Lower magnifications shrink the Unify GUI window to take up less screen space.
- **Data folder**
  - Allows you to change the location of Unify's main data folder (where Libraries/samples and other disk-space hogging content are kept)
  - See under “Data folder” below
- **Patch database**

    - Unify's *patch database* is basically an “index” to all the patch files under the *Libraries* folder.
    - See under “Patch database” below
- **Show Middle C as** menu
    - MIDI note number 60, aka “Middle C”, is called either “C3” (Yamaha convention) or “C4” (Roland convention)
    - This menu allows you to choose whichever you're most comfortable with
    - Your choice affects how note-names are displayed throughout the Unify GUI
- **Velocity Curves**
    - You can define a different velocity-response curve for each MIDI channel (1-16)
    - See under “Velocity Curves” below
- **MIDI Layers** checkbox
    - Allows you to choose whether you prefer to see MIDI layers displayed with full detail or collapsed
    - If you're creating your own BPM patches, you'll probably prefer to see full detail by default
    - If you mainly enjoy playing pre-built patches, you might prefer to see them collapsed
- **Pan Method** menu
    - This is a somewhat-experimental feature for adjusting how Unify's layer-pan controls work
    - See under “Pan Methods” below

Data folder

- The current path to the data folder is displayed on the right
- Click the “Open…” button to quickly open the folder in Explorer (Windows) or Finder (Mac)
- Click the “Change…” button to choose a different location

Patch database

Unify's *patch database* is basically an “index” to all the patch files under the *Libraries* folder. If you should need to delete, add, or rename any of these files, you need to **rebuild the database**

Velocity Curves

Pan Methods

Adding new Patch Libraries

Keep watching the PlugInGuru YouTube channel. From time to time, John “Skippy” Lehmkuhl will announce and demonstrate new patch-libraries for Unify, as they become available.

Adding new patch libraries to Unify involves basically the same steps you followed to purchase and install Unify itself:

1. **Purchase** online at PlugInGuru.com
2. **Download** as a *.guru* file
3. **Install**: drag/drop *.guru* file into the Unify GUI window
4. **Activate** by entering the *license key* from your purchase receipt
5. **Play!**

Each of these major steps breaks down into smaller steps, as described in the sections below.

Purchase your new Library

Go to PlugInGuru.com, and browse the selection of libraries there. When you find one you'd like to buy, log in using your username and password, and click the *Add to Cart* button, then follow the instructions to complete your purchase.

Shortly after purchase, you'll receive a receipt by email, which contains a *download link* (for step 2) and a *license key* (for step 4).

Download the .guru file

Click on the *download link* in your receipt email to download your new library, in the form of a single file whose name is the same as the library you purchased, ending with “.guru”. (You might not actually see the “.guru” extension, depending on how your operating system's settings.)

Once your download is complete, locate the downloaded file in your computer's *Downloads* folder.

Install your new library

Run Unify (preferably the stand-alone app version, but the plug-in versions will work also), and drag/drop the *UnifyStandardLibrary.guru* file icon into the Unify window. Unify will ask if you want to install the new library; click OK to start. A small window will pop up, showing the progress of the installation, and when it's done, this will be replaced by another small window announcing that your new library is now installed; again click OK to dismiss this second window.

If you run out of disk space

Unify's libraries can be quite large—a Gigabyte or more—because nearly all of them contain lots of *samples*. If the disk-drive containing Unify's main data folder does not have enough free space to install your new library, you'll see a warning window instead of the progress window. Before you can install the library, you will have to do one of two things before you can try the drag/drop again:

1. Delete some files on the drive, to free up enough space to install the library, OR
2. Move Unify's main data folder to another drive.

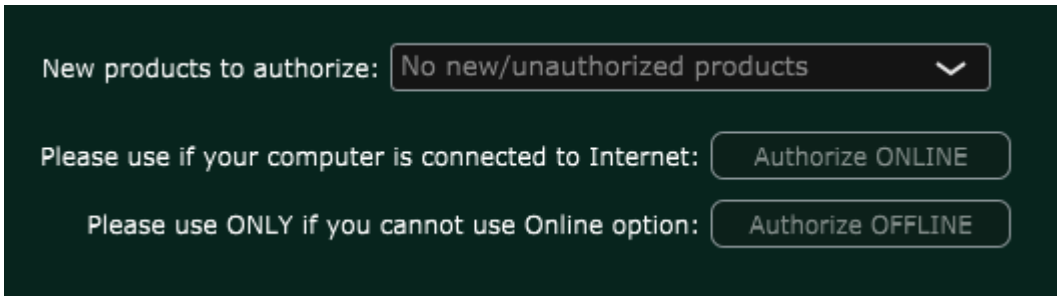
The procedure for option 2 is described in the section on Keeping Unify's data in a non-standard location.

Activate with your license key

As soon as you finish installing a new library, Unify's Licensing View will open automatically, and the newly-installed library will automatically be selected in the “New products to authorize” menu.

At this point, go back to your receipt email and locate the *License Key* for your new patch library. A License Key is a string of 32 letters and numbers (only the digits 0-9 and letters a-f will appear), e.g. `e4cdea654922e3a6320c3984631fe52c`.

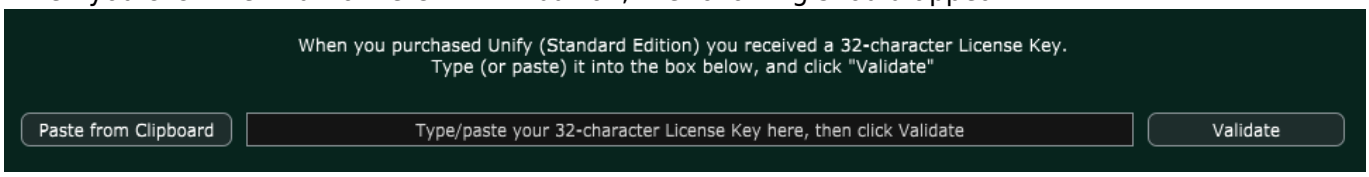
The *Licensing view* will look something something like this:



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 should appear:

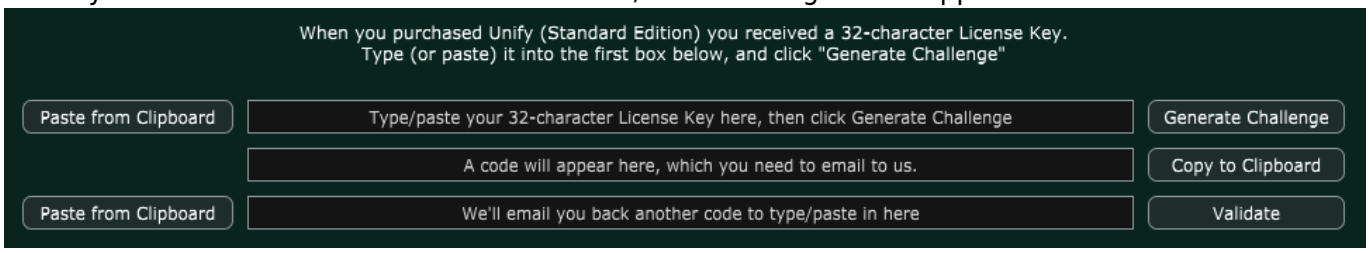


- Locate your 32-character License Key, select it, and use Ctrl-C (Windows) or Cmd-C (Mac) to copy it to your computer's clipboard.
- Click the “Paste from Clipboard” button in the Unify GUI, to put the License Key into the edit box, then click the “Validate” button.
- If all goes well, within a few seconds you'll be rewarded with a message in green text, saying “Valid license code accepted for” followed by the name of your new library.

- If not, whatever message does appear should guide you in what to do next. Contact UnifySupport@PlugInGuru.com immediately if you need help.

Offline authorization

When you click the “Authorize OFFLINE” button, the following should 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 for you (e.g. copy/paste into a document on a USB drive), you need to get this code into a new email message and send it to 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, re-start 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 all goes well, you'll immediately be rewarded with a message in green text, saying “Valid license code accepted for” followed by the name of your new library.
- If not, whatever message does appear should guide you in what to do next. Contact UnifySupport@PlugInGuru.com immediately if you need help.

Play!

As soon as you have completed the authorization process, Unify will automatically rebuild its patch database. This may take up to a minute or so (depending on the speed of your computer's disk drive), but when it's done, you'll be able to see and select the new library's patches in the patch browser (sidebar).

One of the first things you should do is to select the name of your new library from the library-select menu at the top of the patch browser (where it says “All Libraries”). This will show only the new library's patches in the chooser area below, so you can audition them one at a time.

Sharing Unify patches

This page is **under construction**. There are more details to add—coming soon!

Have a look at Creating patch libraries for Unify as well. Even if you're not thinking of making a full Unify patch library, many of the issues discussed there also apply to sharing individual patches.

1. Attach zip archives to your posts (even if only posting one file), as some users may not be able to download files of other types.
2. Include a brief description in your post, and ideally also in a READ ME file included in your zip archive.
3. ALWAYS save with Library = “User Library”.
4. List all third-party plug-ins your patch (or layer preset, etc.) requires, and which formats (VST/VST3/AU) you used.
5. VST is the preferred choice to ensure patches created on Mac can be used on Windows, and vice versa.

Regarding #3: Unify gets the library name for each patch from inside the patch file, NOT the folder the file is in. So if you save a patch with the Library field set to “Unify Standard Library”, it will show up as being part of that library even if the file itself is inside some other folder. This is guaranteed to confuse people! The nice thing about setting Library = “User Library” is that every Unify user will normally have a “User Library” folder, and if they put your patch (.unify file) in there, the library name which shows up in Unify will be the same as the folder they put it into—no surprises.

Sine Wave Synth

Unify's built-in **Sine Wave Synth** is about the simplest synthesizer plug-in possible–so simple, in fact, that it *has no GUI of its own*. It plays sine-wave tones in response to MIDI note-events. It has no envelope, but it does respond to MIDI note-velocity (strike keys harder for louder notes) and pitch-bend (range ±2 semitones).

In future versions of Unify, *Sine Wave Synth* might not be quite so simple…

Guru Sampler

Several users have asked if it's possible to import your own samples into *Guru Sampler*. Yes, it's possible, but the process is NOT simple, and **WE DO NOT PROVIDE SUPPORT** for it. Click here for more information.

If all you want is a SoundFont player, get SForzando by Plogue Art et Technologie. It's excellent, and FREE for both Mac and Windows.

Guru Sampler is Unify's primary built-in instrument. It's a MIDI-controlled, mono/polyphonic sample playback engine with support for velocity-switching and various kinds of randomized playback.

Guru Sampler's graphical user interface (GUI) window presents all parameters in a single view:

The single view is partitioned into ten distinct sections, which are discussed one by one below.

Master section

The controls in the Master section affect how *Guru Sampler* responds to MIDI note-events by allocating voices, choosing samples, and setting note pitch:

- The **first pop-up menu** selects how voices are allocated in response to MIDI note-on events:
  - *Polyphonic:* Each MIDI note is allocated its own voice, up to a maximum set by the **Voices** knob
  - *Mono/Retrig:* A single voice is used, and the sample and envelopes are re-triggered for each note-on event.
  - *Mono/Legato:* A single voice is used, and the sample and envelopes are triggered only for the first note in a *legato* sequence.
- The **second pop-up menu** selects whether samples are pitched according to the MIDI note-number:
  - *Pitched:* the pitch of each sample is adjusted so notes play in tune
  - *Unpitched:* each sample is played at its *original* pitch
- The **Voices knob** is active only in polyphonic mode. It sets the maximum number of voices which may sound at once, from 1 to 64.
- The **Level** knob controls the overall instrument volume.
- The **Shift** knob controls how *Guru Sampler* selects samples for each note:
  - When the knob is in the center (default), the sample-map is obeyed exactly
  - Turning the knob to the right (clockwise) causes progressively lower samples to be chosen for each note, which are then played back faster to bring them up to pitch, resulting in a brighter sound.
  - Turning the knob to the left (counter-clockwise) causes progressively higher samples to be chosen for each note, which are then played back slower to bring them down to pitch, resulting in a darker sound.
- The **Transpose** knob applies an integer offset to MIDI note-numbers. The effect is the same as adjusting the transpose setting for the Instrument layer.
- The **Coarse** and **Fine** tuning knobs offset the whole instrument's tuning by semitones (coarse) or cents (fine).
- The **Glide** knob (aka *Portamento*) is only active in mono/legato mode. It sets the rate (in seconds per octave) at which the pitch glides from one note to the next in *legato* passages.

Amplitude Envelope section

The **Attack**, **Decay**, **Sustain**, and **Release** knobs control the shape of the “ADSR” amplitude-envelope applied to each note. Note that many samples—especially those recorded from acoustic instruments—already have a natural volume contour, which the ADSR curve modifies. The default shape—zero attack and decay times with 100% sustain—allows the natural volume contour of each sample to be heard without any modification.

The **Vel** (velocity sensitivity) knob controls the extent to which note volume is affected by MIDI key velocity. At 100%, the response is extremely dynamic; at 0%, all notes play at the same volume

regardless of key velocity.

Sample selection controls

The three pop-up menus in the **sample selection controls** area allow you to select which sample-map *Guru Sampler* uses:

- The **leftmost menu** allows you to select whether *Guru Sampler* looks at the *Samples* folder of an available Unify *library* or in the “global” *Samples* folder at the top-level of the Unify data folder.
- The **middle menu** allows you to select a sub-folder within the selected *Samples* folder.
- The **rightmost menu** selects which sample-map is chosen from the selected sub-folder.

No sample-map will be loaded until a valid item is selected in all three menus.

Just to the right of the rightmost menu are two **buttons** marked “-” and “+”:

- The **+ button** selects the *next* sample-map from the list in the rightmost menu
- The **- button** selects the *previous* sample-map

Sample Start and Randomize sliders

The two sliders immediately below the row of sample-select menus apply a variable offset from the start of each sample:

- The **Sample Start** slider represents the entire length of the sample.
    - If the “thumb” (green dot) is all the way to the left (default), each sample begins playing back at the beginning, so you will hear the natural attack.
    - Moving the thumb to the right causes playback to begin at the corresponding fraction of the length of the sample, thus bypassing part or all of the natural attack.
- The **Randomize** slider applies a random offset to the sample-start position:
    - If the “thumb” is all the way to the left (default), the sample-start position will be the same for all notes.
    - Moving the thumb to the right applies increasingly larger random offsets to the start position.
    - If the thumb is all the way to the right, the sample-start could be chosen anywhere along the entire length of the sample.

Per-voice Filter section

Like a regular synthesizer, each of *Guru Sampler*'s voices features a resonant filter after the sample oscillator, whose *cutoff* parameter can be modulated by an ADSR envelope which is triggered at the same time as the amplitude envelope, but is separate from it.

- The **left pop-up menu** selects the type of filter used:
  - *OFF:* no filter is applied
  - *Lowpass:* low-pass filter
  - *Highpass:* high-pass filter
  - *BandPass-1:* band-pass filter, type I
  - *BandPass-2:* band-pass filter, type II
  - *Notch:* notch (band-reject) filter
- The **right pop-up menu** selects the number of cascaded filter stages, which in turn affects the steepness of the filter roll-off:
  - *12 dB*: one (two-pole) filter stage, 12 dB/octave roll-off
  - *24 dB*: two stages, 24 dB/octave roll-off
  - *36 dB*: three stages, 36 dB/octave roll-off
  - *48 dB*: four stages, 48 dB/octave roll-off

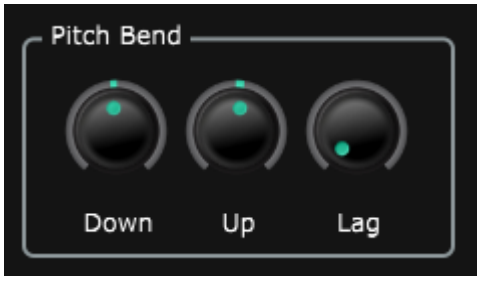
- The **Cutoff** knob adjusts the filter cutoff frequency. As this is a key-tracking filter, the numeric frequency in Hertz displayed while adjusting the knob is for Middle C (MIDI note 60).
- The **Res** (resonance) knob sets the filter's resonance in approximate dB
- The **Gain** knob controls overall filter gain:
  - Reduce when using very high values of resonance
  - Increase when e.g. high-pass filter output would otherwise be too quiet
- The **Vel** (velocity sensitivity) knob adjusts the amount by which the cutoff frequency is adjusted based on MIDI key velocity:
  - In the center position (default), cutoff is unaffected by velocity
  - Turning the knob to the right causes notes which are struck harder (higher velocity) to sound brighter
  - Turning to the left yields the reverse effect: the harder notes are struck, the darker they sound.
- The **Key Trk** (key tracking) knob adjusts how faithfully the filter cutoff tracks note pitch:
  - When the knob is all the way to the right, cutoff tracks pitch exactly.
  - When it is all the way to the left, there is no key tracking at all.
  - The default position of 60% yields a good compromise, with higher notes being progressively darker to avoid sounding too strident.

Filter Envelope section

Each voice features an “ADSR” envelope generator to modulate the filter cutoff, which is triggered by MIDI note-on events, simultaneously with the amplitude envelope generator. Note the controls in the Filter Envelope section are only active if the first pop-up menu in the Filter section is NOT set to “OFF”.

- The **Attack**, **Decay**, **Sustain**, and **Release** knobs control the shape of the ADSR envelope.
- The **Env Amt** (envelope amount) controls the modulation strength. When the knob is all the way to the left, the envelope has no effect at all.
- The **Vel** (velocity sensitivity) knob controls the extent to which the modulation strength is affected by MIDI key velocity. At 100%, the response is extremely dynamic; at 0%, key velocity has no effect on cutoff modulation.

Pitch Bend section



The Pitch Bend controls allow you to adjust how the *Guru Sampler* responds to MIDI pitch bend.

- The **Down** and **Up** knobs control how far (in semitones) the pitch will bend in each direction.
- The **Lag** knob applies low-pass filtering to the pitch-bend input, with the result that note pitch lags behind the movement of the MIDI pitch-wheel.
    - When the knob is all the way to the left (default), there is no lag.
    - Moving the knob to the right yields progressively greater lag.

The Down and Up knobs are bidirectional:

- In the center position (default), the pitch wheel has no effect
- Turning the knob to the right yields increasingly positive effect.
- Turning to the left yields increasingly negative effect.

Typically, Down will be set to a negative value (default -2 semitones) and Up to an equal positive value (default +2 semitones), but you can vary these settings for creative effect.

LFO section

Guru Sampler has a single global (not per-voice) low-frequency oscillator which can affect both pitch and filter-cutoff.

- The **left pop-up menu** selects among various LFO waveforms:
  - *Sine*, *Triangle*, and *Square* are self-explanatory
  - *Saw Up* is a sawtooth waveform which starts at 0 and ramps up to 100% on each cycle
  - *Saw Dn* (down) is a sawtooth which starts at 100% and ramps down to 0 on each cycle
  - *S/Hold* (sample-and-hold) is a stepwise random waveform created by running white-noise signal through a sample/hold function.
  - *Random* is a smoothed version of the sample-and-hold waveform.
- The **right pop-up menu** selects the LFO waveform polarity:
  - *Bi* (bidirectional) means the waveform modulates symmetrically above and below zero
  - *Uni* (uni-directional) means the waveform has only positive excursions; it never goes below zero.
- The **Rate** knob sets the LFO frequency
- The **Pitch** knob sets the amount by which the LFO modulates pitch
- The **Cutoff** knob sets the amount by which the LFO modulates filter cutoff

The Pitch, and Cutoff knobs are bidirectional:

- In the center position (default), the LFO has no effect
- Turning the knob to the right yields increasingly positive effect.
- Turning to the left yields increasingly negative effect.

Mod Wheel section

The controls in the **Mod Wheel** section affect how *Guru Sampler* responds to MIDI CC#1 (mod wheel).

- The **Pitch** knob adjusts how much the mod wheel affects note pitch
- The **Cutoff** knob adjusts how much the mod wheel affects filter cutoff
- The **LFO/Pitch** knob adjusts how much the mod wheel affects how strongly the LFO modulates

pitch.
- The **LFO/Cutoff** knob adjusts how much the mod wheel affects how strongly the LFO modulates filter cutoff.
- The **Lag** knob applies a variable amount of low-pass filtering to the mod-wheel input, with the result that the response lags behind the actual position of the MIDI mod-wheel.

The Pitch, Cutoff, LFO/Pitch, and LFO/Cutoff knobs are all bidirectional:

- In the center position (default), the mod wheel has no effect
- Turning the knob to the right yields increasingly positive effect.
- Turning to the left yields increasingly negative effect.

Importing your own samples into Guru Sampler

We DO NOT offer end-user support for what's described here.

The forthcoming *Unify Pro* will include a full graphical user interface (GUI) for importing and mapping your own samples. What's described here is the complex, manual process we use now, in all its nasty glory.

If you want to try this, do so at your own risk!

Guru Sampler uses a VERY LIMITED variant of the "SFZ" format. In fact, it's better to say that Guru Sampler's format is “inspired by” the SFZ format, because our .sfz files aren't even recognized by standard SFZ players like Sforzando.

Each Guru Sampler .sfz file is a simple text file consisting of a sequence of Groups, each of which defines a region of the keyboard. Each Group contains a <group> header line, followed by one or more <region> lines, each of which maps one sample file to a specific velocity range within the key group.

Downloading the example files

Click on these links to download two SFZ examples:

- single sample, same as *PAD-Sonitarium E4* from *Signs of Life*: single-sample.zip
- 11 zones, 3 velocity layers, using samples extracted from the *mda-ePiano* VST:

mda-epiano.zip

In each case, the zip archive expands to a single SFZ file and a folder called *samples* containing WAV files.

Structure of the SFZ file

Let's look at the first four lines of mda-ePiano.sfz:

```
<group> lokey=0 hikey=39 pitch_keycenter=36
  <region> lovel = 0 hivel = 47 loop_mode=loop_sustain loop_start=4076
sample=samples/ePiano01.wav
  <region> lovel = 48 hivel = 79 loop_mode=loop_sustain loop_start=2868
sample=samples/ePiano02.wav
  <region> lovel = 80 hivel = 127 loop_mode=loop_sustain loop_start=11918
sample=samples/ePiano03.wav
```

The <group> line says we are mapping a group of samples to the MIDI note-number range 0 through 39 (inclusive), and that the pitch of each sample is that of MIDI note-number 36.

The three <region> lines define how three separate samples (ePiano01.wav, ePiano02.wav, ePiano03.wav) are assigned to three MIDI velocity-ranges.

- *ePiano01.wav* is for note-on velocities from 0 to 47. It loops through the sustain and release periods, starting from sample-index 4076 and continuing to the end.
- *ePiano02.wav* is for note-on velocities from 48 to 79. It loops through the sustain and release periods, starting from sample-index 2868 and continuing to the end.
- *ePiano03.wav* is for note-on velocities from 80 to 127. It loops through the sustain and release periods, starting from sample-index 11918 and continuing to the end.

Some things to note:

- Typically, the <group> lokey/hikey ranges will be non-overlapping and span the range 0-127, but this is neither required nor enforced by Guru Sampler.
- Typically, the <region> lovel/hivel ranges will be non-overlapping and span the range 0-127 (technically 1-127 because velocity 0 is not used for note-on), but again, not required.
- Anytime you have overlap, where two or more separate sample files are assigned to the same key/velocity, Guru Sampler will choose randomly among them for each note, yielding a kind of “poor man's round-robin” effect.
- Guru Sampler is NOT a “streaming” sampler (yet). All samples are loaded fully into RAM. It is up to you to limit the RAM footprint, which will be 4/3 the size of 24-bit WAV files, or 2x the size of 16-bit WAV files, because 4 bytes/sample are used.
- You CAN use AIFF or other standard formats; you don't have to use WAV.

Each <group> line may contain the following tags (default values in parentheses):

- lokey (0), hikey (127) : MIDI key-number limits
- pitch_keycenter (64) : MIDI key-number on which the sample's assumed pitch is based

Each <region> line may contain the following tags (default values in parentheses):

- lovel (0), hivel (127) : MIDI velocity limits
- loop_mode : the “value” is ignored; if present the sample will loop on playback
- loop_start (0), loop_end (#samples - 1) : 0-bases indices of start/end of loop region
- tune (0.0) : amount, in cents (decimals OK) by which the sample pitch must be raised (positive value) or lowered (negative) to exactly match the pitch_keycenter value
- sample=<path to sample file> : identifies the sample file
    - The path is relative to the folder containing the .sfz file
    - The path MAY contain embedded spaces, and ends at the end of the text line

Note you do NOT have to put new sample-sets into Samples folders under Unify's Libraries folder. You can add a folder called Samples directly in the main PlugInGuru/Unify folder, (sibling to the Libraries folder) and create sub-folders in there, containing your SFZ files (more than one if you wish) To open them in Guru Sampler, choose “<Samples Folder>” in the leftmost menu, then the name of your sub-folder in the middle menu, and finally the name of the SFZ file you want in the rightmost menu.

A simpler example

When all you need to do is map one sample across the entire keyboard, the SFZ is much simpler, requiring only one `<group>` line and one `<region>` line. Here is *PAD-Sonitarium E4.sfz* in its entirety:

```
<group> lokey=0 hikey=127 pitch_keycenter=76
<region> end=1066811 loop_mode=loop_continuous loop_start=241735
loop_end=1066810 lovel=0 hivel=127 tune=0 sample=samples/PAD-Sonitarium
E4.wav
```

The `<group>` line has

- `lokey=0 hikey=127`, meaning the entire MIDI key-number range. (You could actually omit these entries entirely, since these are the default values for `lokey` and `hikey`.)
- `pitch_keycenter=76`. This is the really important part, which indicates that the associated sample will sound at its natural pitch (not sped up or slowed down) at MIDI note number 76 (aka E4).

The `<region>` line has

- `end=1066811` indicating the sample-index of the last sample. (This could have been omitted, because the sample file is exactly 1,066,812 samples long, and `end` defaults to the index of the last sample.)
- `loop_mode=`*something* (The `=loop_continuous` could actually be anything; all that matters is that the `loop_mode` part is there.)
- The `loop_start` and `loop_end` items are the important part. They indicate the sample-indices of the first and last sample of the segment which loops indefinitely while a key is held down.
- `lovel=0 hivel=127` indicates that this sample is played for the entire MIDI note-velocity range. These items could be omitted, as 0 and 127 are the defaults, but we include them here to indicate that you could also set e.g. `hivel=80` to respond only to velocities 0–80.
- `tune=0` means the pitch of this sample is not adjusted up or down. To tune the sample UP by 10 cents, for example, you would say `tune=-10`. (Negative values push the pitch UP, positive values push it DOWN; don't ask why.)
- `sample=samples/PAD-Sonitarium E4.wav` identifies the sample file. It must always be the *last item on the line*.

Converting from other sampler formats

It would be wonderful if you could get a program which would magically convert key-mapped sample sets for other samplers (like *ESX24* or *Kontakt*) to the SFZ format used by *Guru Sampler*. Unfortunately, there is no such thing.

The best available option is Translator by Chicken Systems. The full version costs $149, but you're only interested in converting to SFZ, you can buy the “Special Edition” for $79. It can indeed convert almost any sampler format to what it calls “Cakewalk SFZ”, BUT

- These files require manual editing, because they're not quite the same as what *Guru Sampler* needs

- In particular, the files will contain multiple redundant copies of several tag items.

Translator is **no magic bullet**. It's an exceptionally quirky and confusing program. If you use it, steel yourself for frustration, confusion, and regret about how much you paid for it. Its user-interface is not at all intuitive, especially the “bulk conversion” features (which can easily leave you wondering why you suddenly have *multiple copies* of *many large WAV files* on your disk).

Are you SURE you want to try this?

Seriously, unless you're a bit of a masochist, save yourself the money, time, and trouble, and just wait for *Unify Pro*. It will have all the easy, drag-and-drop, graphical-editing loveliness Unify users have asked for.

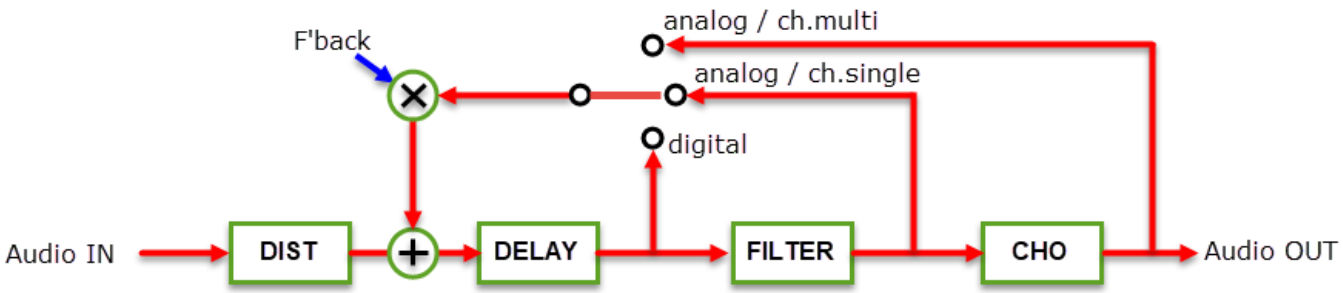
And remember what we said at the start: **WE DO NOT PROVIDE SUPPORT FOR THESE PROCEDURES**. You're on your own.

Omega Delay

Omega Delay is Unify's built-in stereo delay audio effect, featuring independent Left and Right delay lines, free and BPM-synced timing (with adjustable drift offset), multi-mode filters in the feedback path, and integrated Distortion and Chorus effects.

Signal flow

The signal flow in *Omega Delay* is shown in the following diagram. Input audio passes first through the Distortion block and then through the Delay, Chorus, and Filter blocks. The feedback signal may be tapped from the output of the latter three blocks as shown. All signal paths are stereo.



The *Omega Delay* GUI is divided into four sections, described individually below. By starting with the *Distortion* section and proceeding clockwise around the GUI, we essentially follow the signal path as shown above.

Distortion section

The Distortion block consists of two identical wave-shaping blocks, i.e., the Left and Right signals are processed identically.

- The menu at the top allows choosing the **curve shape**
  - **OFF** disables the Distortion entirely

    - **Exponential** and **Power Curve** are two variants of a convex curve
    - **Hard Clip** provides a variable linear gain segment with clipping at maximum amplitude
    - **Soft Clip** provides a high-gain segment and a low-gain segment, with a smooth “knee” between them
    - **Quantize** yields an adjustable stair-step curve
- For all curve shapes, the **Strength** knob adjusts the amount of distortion, ranging from linear response (no distortion) when the knob is all the way to the left, to extreme distortion when all the way to the right.
- The **wave-shape graph** helps you visualize what the *Strength* knob is doing to the shape
    - The horizontal axis represents input signal amplitude (absolute value, above or below zero)
    - The vertical axis represents the corresponding output signal amplitude
- The **Drive** knob adjusts the *pre-distortion* signal gain (before wave-shaping)
- The **Gain** knob adjusts the *post-distortion* signal gain (after wave-shaping)

Delay section

The Delay block is a basic stereo digital delay, with delay times up to 2000 milliseconds. The pop-up menu at the top allows choosing how the two delay-time knobs, **D.Time L** and **D.Time R** are interpreted.

- In **No Sync** mode, delay times are adjustable from 0 to 2000 ms, in milliseconds
- In **BPM Sync** mode, delay times are defined relative to the current beat time
    - 1/4 (quarter-note) is the default
    - 1/64 is minimum, 1/1 (one measure) is maximum
    - Suffix T means “triplet” (times 2/3), D means “duplet” (times 3/2)

In *BPM Sync* mode, the **Drift** knob is active, and can be used to adjust the Left and Right delay times symmetrically away from their exact beat-relative values.

- Turning the knob to the **right** makes the Right delay time longer, and the Left shorter
- Turning to the **left** makes the Left delay time longer, and the Right shorter

The two **Fback** (feedback) knobs control the amount of feedback in the Left and Right delay lines individually.

The **Dry/Wet** knob controls the mix between the unprocessed “dry” signal (*Audio IN* in the diagram above) and the processed “wet” signal (*Audio OUT* in the diagram) that gets passed on the the output of *Omega Delay*. (For clarity, this is not shown in the signal-flow diagram.)

Feedback Filter section

The *Filter* block is a single (stereo) two-pole recursive digital filter with adjustable **Frequency**, **Q** (resonance), and **Gain**.

The stack of three controls (two menus, one button) at the left-hand side of this section requires explanation.

- The **top menu** lets you choose the *filter type*:

    - *Low-pass*, *high-pass*, *band-pass*, and *notch* (band-reject) are standard filter types
    - *Low-shelf*, *high-shelf* and *peak* are filter types normally found in EQ units. These allow emphasizing/de-emphasizing a specific frequency range, while passing the rest through unchanged (unity gain).
- The **middle menu** defines how the feedback signal is filtered over time:
    - **Analog** feeds back the output of the Filter block, so echoes become progressively more strongly filtered, as in a typical analog delay system.
    - **Digital** feeds back the output of the Delay block, so all echoes are filtered (and optionally chorused) the same amount.

Chorus section

Omega Delay also features a stereo chorus effect between the Delay and Filter blocks. The four knobs are fairly conventional for a chorus effect, which uses a quadrature LFO (low-frequency oscillator) to modulate the delay-times of two very short delay-lines, with a 90-degree phase difference between Left and Right.

- **Rate** controls the frequency of the modulation LFO
- **Depth** controls the modulation depth
- **Feedback** provides adjustable feedback, for more extreme creative effects (leave at zero for regular chorus)
- **Mix** controls the mix between the “dry” unprocessed signal and the “wet” chorused signal.

The pop-up menu allows you to choose one of three “modes” or signal routing options as follows, with reference to the signal-path diagram above.

1. **OFF** disables the chorus entirely
2. **Single** selects the output of the Filter block as the feedback signal, so the chorus effect is applied only to the outgoing signal.
3. **Multi** selects the output of the Chorus block as the feedback signal, with the result that successive echoes pass through the Chorus multiple times.

Finally, the **Kill FB** button instantly clears both Left and Right delay lines. This can be useful if you have used a lot of gain, distortion, and/or feedback, driving the delay lines into oscillation (repeating signal even when there is no more input). (This button technically belongs in the *Delay* section, but is here because it fits better in the layout.)

NoizBox (distortion effect)

NoizBox is a creative distortion effect consisting of an *input filter*, a *nonlinear wave-shaper*, and an *output filter*, with a *Dry/Wet Mix* control to adjust the balance between the “dry” unprocessed signal and the “wet” distorted signal at the output.

The GUI consists of three sections, described individually below.

PreFilter section

The **PreFilter** section of the GUI represents the input (pre-distortion) filter. This is actually a chain or cascade of up to four identical, two-pole recursive digital filters.

- The **left menu** allows selecting the filter type:
  - **OFF** disables the pre-distortion filter block entirely
  - **LP**: low-pass
  - **HP**: high-pass
  - **BP**: band-pass
  - **BR**: band-reject (aka “notch”)
- The **right menu** allows selecting the number of stages in the filter chain:
  - **12 dB**: one stage, yielding 12 dB/octave roll-off
  - **24 dB**: two stages, yielding 24 dB/octave
  - **36 dB**: three stages, 36 dB/octave
  - **48 dB**: four stages, 48 dB/octave
- The three **knobs** adjust the frequency response and gain of all active stages:
  - The **Freq** knob adjusts the filter's characteristic frequency (e.g. cutoff or center frequency)
  - The **Q** knob adjusts the amount of *resonance* (aka “emphasis”) near the characteristic frequency
  - The **Drive** knob adjusts the filter's *gain*. It's called “drive” because it affects how hard the nonlinear waveshaping block is driven.

PostFilter section

The **PostFilter** section of the GUI represents the output (post-distortion) filter. Like the pre-filter, this is also a chain of up to four cascaded filter sections, and the controls are the same.

The **Mix** knob adjusts the balance of unprocessed (“dry”) and processed (“wet”) signals in the final

output.

- When the knob is all the way to the left, only the dry signal is heard
- When the knob is all the way to the right, only the wet signal is heard
- Intermediate positions set the mix of dry and wet.

Waveshaper (graph) section

The **graph** section in the middle of the *NoizBox* GUI represents the nonlinear wave-shaper block, which is the heart of the distortion unit.

The graph itself represents the positive half of the distortion response curve, i.e., how the wave-shaper responds to positive-valued input samples. The negative half of the curve is identically shaped, but reversed in both X and Y directions.

Clicking on the graph pops up an editor window like this:

The **pop-up menu at the top left** allows you to select the basic curve shape:

- **OFF** disables the Distortion entirely
- **Exponential** and **Power Curve** are two variants of a convex curve

- **Hard Clip** provides a variable linear gain segment with clipping at maximum amplitude
- **Soft Clip** provides a high-gain segment and a low-gain segment, with a smooth “knee” between them
- **Quantize** yields an adjustable stair-step curve
- **Custom** yields a curve you can edit into just about any shape–see below.

Immediately to the left of the large graph box is a vertical **Strength slider**. For the exponential, hard clip, soft clip, and quantize shapes, this adjusts the amount of distortion, ranging from linear response (no distortion) when the “thumb” (green dot) is all the way to the bottom, to extreme distortion when all the way to the top.

Editing Custom distortion curves

When you select *Custom* from the curve-shape menu, the large graph area becomes an interactive *curve editor* which works basically the same as the editors for MIDI-velocity curves in Unify.

- Drag either endpoint (circles) up or down to change the output range
    - You can drag the right-hand endpoint *below* the left one for an inverted response (increasing input signal results in *decreased* output signal, aka “fold-over”)
- Drag up/down on the line between the endpoints to adjust the shape of the curve (convex or concave)
- *Double-click* between endpoints to add a new “split point”
    - This splits the curve segment in two
    - Split points can be dragged up/down and left/right
    - Individual curve segments can be dragged up/down to adjust shape
- Double-click any split point to delete it
    - This merges two segments back into one

Resetting, Saving and Loading curve presets

NoizBox allows you to save distortion curve-shapes individually as *Distortion Curve Presets*. Note all of the different curve-types can be saved this way, not just “custom” curves.

- Click the **Reset** button to reset the distortion curve to a straight line (no distortion at all)
- Click the **Save…** button to save the current curve shape to a new preset file
    - A standard file-save dialog will pop up, allowing you to choose the file name and location
    - It is recommended to stick with the default location, which is a folder called *Distortion Curves* within the main Unify *Presets* folder
- Click the **Load…** button to re-load a previously saved preset
    - A standard file-open dialog will pop up, with the standard *Distortion Curves* folder already selected

Creating patch libraries for Unify

This page is **under construction**. There are many more details to add—coming soon!

Commercial plug-ins

When making libraries for Unify, you have to be very careful about commercial plug-ins:

- If you make a library using only plug-ins that are included with Unify, EVERY Unify owner will be a potential buyer.
- If your library also requires ONE commercial plug-in, your market shrinks to Unify owners who already own that plug-in, plus a few more who might buy it just because of your patches.
- If your library requires TWO commercial plug-ins, you would be able to do some really interesting creative things, but your market will be smaller still.
- If you make a library that uses MANY commercial plug-ins in various combinations, almost no one will buy it, because they won't know how many of your patches will work for them. Those that do buy will be very upset every time they see all the other patches they paid for, but can't use.

Cross-platform considerations

You also have to think about cross-platform (Mac/PC) issues:

- Any plug-ins you use MUST be available for both Mac and PC, IN THE SAME FORMAT, e.g. VST or VST3.
- Unify saves the “state” (all settings) of each plug-in, as a chunk of binary data which the plug-in itself provides. When a Unify patch is re-loaded, it instantiates each plug-in, passing in the appropriate binary data.
- The data formats for VST, VST3, and Audio Units are VERY different, and there is no automatic way to convert one to the other, because the differences are plug-in dependent. If you create a patch using, say, the VST3 version of Spire, someone who only has the VST version will not be able to play it.
- In our experiments, we have found that VST (i.e., VST2) provides the best cross-platform compatibility. Patches created using only VST2 plug-ins on a Mac will play correctly on a PC, and vice versa.
- VST3 also seems to work cross-platform, but many plug-in vendors are still figuring out the details of the VST3 standard (Steinberg has not been very helpful), so VST3 plug-ins are often quirky.

Publishing options

- You can always create a patch-library as just a collection of shared patches, which you can distribute yourself.
- If your intent is to create a paid-for product, you will most likely want to have it *encrypted* and

licensed for individual use. This is possible by publishing through *PlugInGuru.com*. Contact John Lehmkuhl admin@pluginguru.com for details.
- *PlugInGuru.com* is NOT a vanity-publishing service. Third-party libraries must meet John's very high standards to be published under the *PlugInGuru* brand.

Technical stuff

There are a few VERY important details to understand before you start making a patch library for Unify:

1. Every *.unify* file contains a setting for the Library it’s supposed to be in. Just moving the file to a different folder does NOT change this.
    1. This may change in future versions of Unify
2. Unify v1.0.6 or earlier permits saving a file without specifying its Library at all.
    - This was a mistake, and will be corrected in later versions.
    - It results in files with a blank Library setting; this confuses Unify as described below.
3. When you specify a new library name when saving a patch, Unify creates a folder with that name, AND a *Patches* folder inside it, where it saves the patch.
4. Both *Guru Sampler* and *KlangFalter* save references to sample/IR files relative to the main Unify data folder.
    - For your library to be self-contained, your library’s patches to point to your library’s samples
    - If you put your library’s samples into the *Samples* folder beside the *Libraries* folder, you’ll need to re-select them all, as described below.

While on the subject of library names, don't choose something too generic like “Dance vol.1”. Think ahead to when a user has 80 different libraries. They'll need to be able to find yours easily, so name your library's the way you'd name a band, in a way which is somehow unique to you.

Here’s what you need to do when creating a new library:

- Create the first patch (even an INIT patch is OK) and hit **Save**
    - Choose “New Library…” and enter a new library name
        - If you can’t do that, see below
    - Click Save **New…** and make sure you’re saving into the *Patches* folder under *Libraries/<your library name>*
- Create your library’s *Samples* folder (see below)
- Select your library name when saving each new patch you create

If you choose the “New Library…” item in the library list, but Unify does not present a text editor where you can type the new name, it’s because you previously saved some patches without specifying a library at all. This confuses the program.

- In the patch browser’s Libraries menu, choose <none specified>
- Open each patch and Save As with a real library name. If necessary use “User Library”.

Once there are NO more patches with a blank Library field, the <none specified> item should disappear from the Libraries menu. If not:

- Quit and re-start Unify
- Rebuild patch database

All of your own samples should go into a *Samples* folder under *Libraries/<your library name>*. This folder won't be created automatically; you have to make it yourself. If the parent folder (shown below as "Your Library Name") doesn't exist, go back to the top and save one patch there like I said. This will create the parent folder and the *Patches* folder. COPY (don't move) all of your samples into the *Samples* folder, grouped into sub-folders according to the following folder structure:

- Main Unify data folder
  - *Libraries*
    - Your Library Name
      - *Patches*
        - All your patches (*.unify* files) go here
      - *Samples*
        - Subfolder 1
          - SFZ files
          - one or more folders containing WAV, AIF, etc. files
        - Subfolder 2
          - SFZ files
          - one or more folders containing WAV, AIF, etc. files
        - etc.

Note you should not put SFZ files directly inside the Samples folder. They should always be grouped under sub-folders (even if there's only one), or *Guru Sampler* won't be able to load them properly.

The reason you should to COPY, not MOVE your samples is that in all your patches, the saved state every *Guru Sampler* instance is set to look for them in the user *Samples* folder beside the *Libraries* folder. To fix this:

- Open every single one of your patches, one at a time, and re-select the sample, starting by choosing your library name in the first menu
- Save each patch (you can do a quick save by Option/Alt-Clicking the **Save** button)

Only AFTER you have fixed every single sample path in this way, you can finally remove the original samples from the user Samples folder (if you want to; it's harmless to leave them in there).

If you have already created a ton of patches:

- In Finder/Explorer, MOVE the patch files into your library's *Patches* folder.
- In Unify, rebuild patch database. *Note this WON'T change the library names in each patch.*
- Load each patch one at a time, select your new library name, and hit **Save New** to save them in the new library's *Patches* folder.
- Because you moved the patch files in there already, you should be asked each time if you want to overwrite the file; say yes.

These steps should be sufficient to rebuild your library so it's self-contained. If you can, test on another machine by copying ONLY your library's main folder. (Make sure to rebuild the patch database on the test machine after you do this.)

At some point in the future, we'll be adding features to Unify to make some of these procedures a bit more automatic, but that won't happen soon.

Manually unpacking ".guru" files

Some Unify users have found that they are unable to drag/drop *.guru* files into Unify. We still haven't figured out why, and we're planning to add a manual button as an alternative. In the meantime, if you're comfortable working with files, you may want to try unpacking the *.guru* file yourself.

The following instructions are specifically for the *Unify Std Lib 1.0.6 Update.guru* file for Unify v1.0.6, but the basic principle is the same for any *.guru* file.

Make a backup copy of Unify's main data folder

Since you're going to be overwriting many files in Unify's main data folder, we strongly recommend that you **make a backup copy of the entire folder** before starting. To locate the folder:

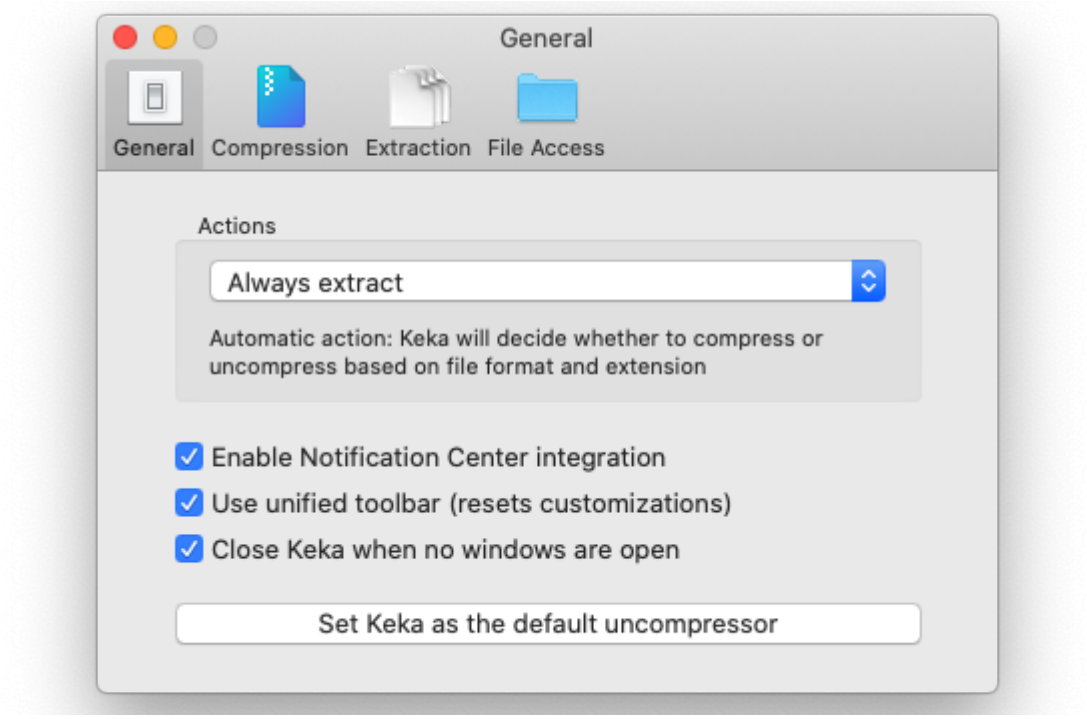
- Open the Unify stand-alone app
- Go to the Settings view
- Click the “Open…” button
- In the Explorer/Finder window which opens, navigate UP one level, to the folder which **contains** Unify's main data folder.
- Either **duplicate** the Unify data folder, or **compress/zip it**.

Install 7-zip (Windows) or Keka (Mac)

Each *.guru* file is actually a standard Zip archive, so IF you can manage to change the *.guru* extension to *.zip*, you can just unzip it using your operating system's built-in unzip utility. **Please don't even try this unless you know exactly what you're doing.** All recent versions of both Windows and MacOS are configured to hide filename extensions by default, and unless you manually enable display of full filenames, you can easily end up with a file that's actually called something like “name.zip.guru” and still can't be opened.

You can avoid all this trouble by installing a decent zip archive utility.

- On Windows, use the excellent (and free) 7-zip
- On Macintosh, use Keka
    - You can download the free installer from the main Keka page
    - Or you can use the “Download from the Mac App Store” to get the “official” one (costs a few dollars).
    - Run the *Keka* app once, open the Preferences, and set the default action to “Always extract”



Unpack the .guru file to a new folder

On Windows, after installing 7-Zip, you can right-click the *.guru* file and choose *7-zip > Extract to "Unify Std Lib 1.0.6 Update/"* to unpack the contents to a new folder with the same name as the *.guru* file.

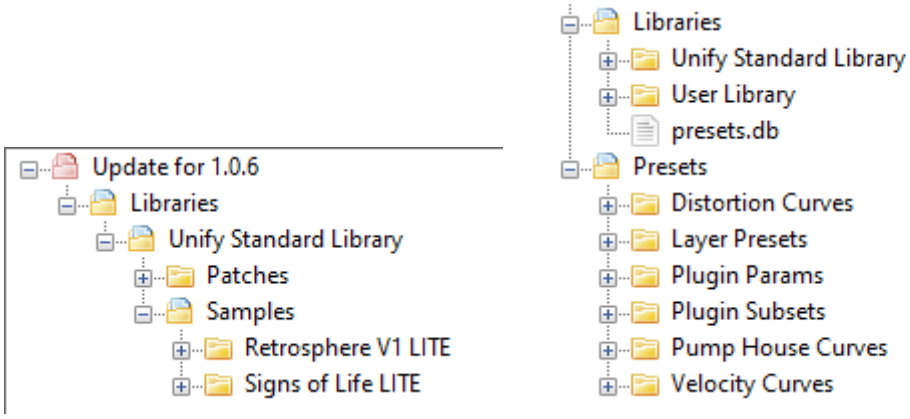
On the Mac, you can right-click the *.guru* file, choose *Open with...*, and select Keka from the list. If you set the default action to "always unpack" as advised above, it should immediately unpack the contents into a new folder with the same name as the *.guru* file.

From this point on, the steps should be the same for Mac or Windows.

Open the new folder AND Unify's data folder

Double-click the new folder you created by unpacking the *.guru* file. Run the Unify stand-alone app, go to the Settings view, and click the "Open..." button to open the main Unify content folder, and then close Unify again.

Now compare the two folders, looking inside the various sub-folders. You'll see that the *.guru* file contents (shown on left below) are a *subset* of the more complete structure in Unify's main data folder (right):



Copy files from the new folder to Unify's data folder

At this point, you can manually copy the new/updated files from the unpacked *.guru* folder to the **corresponding locations** in Unify's main data folder. Remember, **we recommend making a backup of your Unify data folder before starting**.

In almost every case, you'll be copying a new version of a file on top of an older file with the same name, and you'll be prompted to confirm that you want to replace the older file with the newer one. **If you don't see such a prompt, double-check that you're putting the new file into the right place.**

Close folders, clean up, run Unify, rebuild patch database

When you're finished copying files, you should:

- close any open folders
- move the new folder you created when unpacking the *.guru* file, to the Trash
- move the *.guru* file itself to the Trash
- open the Unify stand-alone app, open the patch browser, and click the lightning-bolt icon at the top right corner to rebuild the patch database.

It's a good idea to make sure everything is working as you expect, before emptying the Trash. If you think you may have made a mistake, or forgotten some files, you can drag the folders back out of the Trash and try again.

Once you're truly confident that you've made all necessary changes, and Unify is working as it should, you can go back and delete your backup of the main Unify data folder.