BITWIG STUDIO

USER GUIDE

BITWIG STUDIO 1.1

The content of this user guide is subject to change without notice and does not represent a commitment on the part of Bitwig. Furthermore, Bitwig doesn't take responsibility or liability for errors or inaccuracies that may appear in this user guide. This guide and the software described in this guide are subject to a license agreement and may be used and copied only in terms of this license agreement. No part of this publication may be copied, reproduced, edited or otherwise transmitted or recorded, for any purpose, without prior written permission by Bitwig.

This user guide was written by Dave Linnenbank.

Second Edition, November 2014.

Bitwig GmbH | Schwedter Str. 13 | 10119 Berlin - Germany
contact@bitwig.com | www.bitwig.com

Bitwig Studio is a registered trademark of Bitwig GmbH, registered in the U.S. and other countries. VST is a registered trademark of Steinberg Media Technologies GmbH. ASIO is a registered trademark and software of Steinberg Media Technologies GmbH. Mac OS X, Safari and iTunes are registered trademarks of Apple Inc., registered in the U.S. and other countries. Windows is a registered trademark of Microsoft Corporation in the United States and/or other countries. All other products and company names are trademarks or registered trademarks of their respective holders. Use of them does not imply any affiliation with or endorsement by them. All specifications are subject to change without notice.

© 2014 Bitwig GmbH, Berlin, Germany. All rights reserved.

0. Welcome and Get Ready

Welcome to Bitwig Studio! We are glad you have joined us and are excited to help you create, compose, polish, and perform your music.

The purpose of this document is to walk you through most of Bitwig Studio's functions and show you how to operate the program. The chapters and topics are arranged progressively, with basic concepts appearing first and advanced ideas showing up later. And although this document does not attempt to explain fundamental audio and musical concepts, it is written for users of any stripe who want to use software to make music.

In addition to this document, other resources will be mentioned when appropriate, the program itself offers a fair amount of documentation through tooltips, and you can always visit Bitwig's website [http://bitwig.com] for the latest information. And please share any feedback you have or issues you encounter by visiting our support portal [http://bitwig.com/support].

In this chapter, we will cover how to initially set up Bitwig Studio so that audio is ready to work and any MIDI controller you have is ready to be used. But you will not make sound in this chapter; that is what the rest of this document is for.

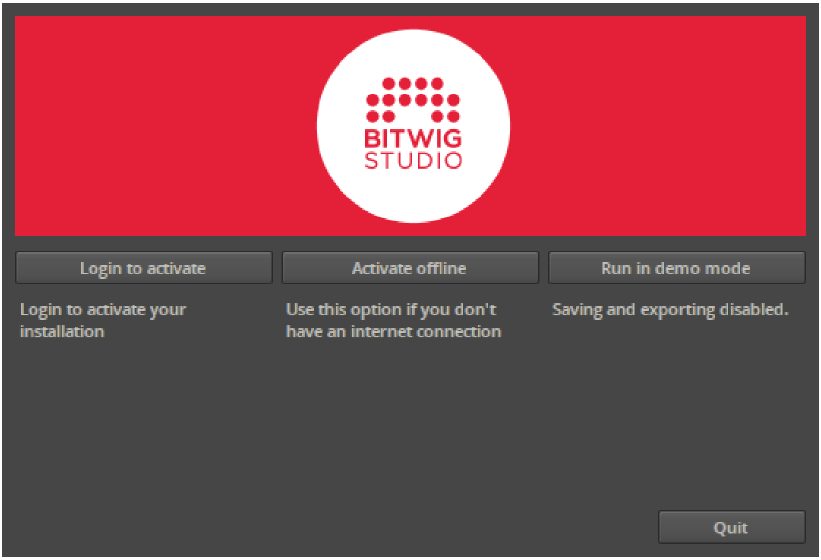
0.1. Opening Bitwig Studio the First Time

Bitwig Studio exists for Windows, Mac OS X, and Linux, and the process of installing it on your computer depends upon the platform you are using. Each installer package contains either an installer that guides you through the process or instructions for copying the application onto your computer.

Once Bitwig Studio is on your machine, it's time to open it up and get ready to start making music. The program itself helps you set things up. This section will guide you through choosing your settings on that first launch and show you how to revisit them later.

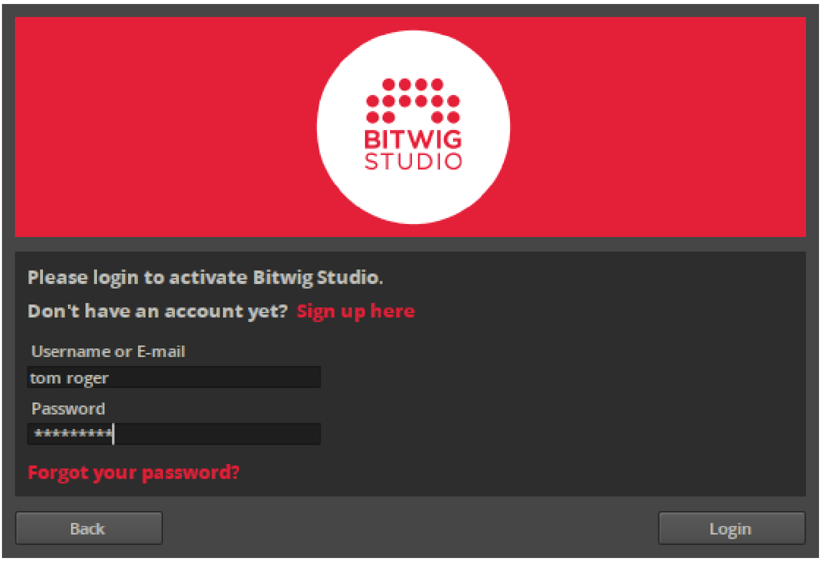
0.1.1. Authorization

After you launch Bitwig Studio for the first time, the program's splash screen will pop up. In a moment, it will disappear, and then a new window will appear.



This window presents us with three clickable buttons:

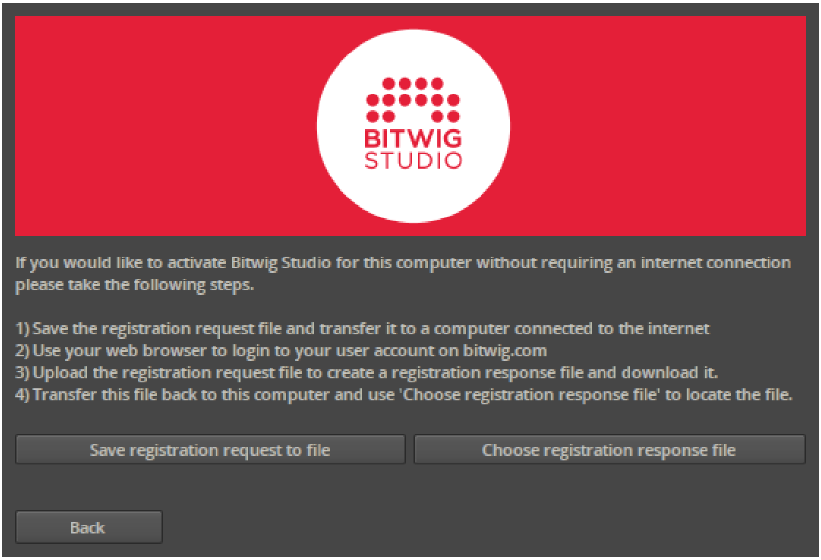
› *Login to activate* will prompt you for the credentials you have created with Bitwig. It requires you to have an internet connection.



If you have not yet created an account with Bitwig, you can do so via a web browser by clicking *Sign up here*. And if you have created an account but cannot remember the password, the *Forgot your password?* link is for you.

Once you know your credentials, enter them in the window and click *Login* to continue.

› *Activate offline* is the right choice if you own Bitwig Studio but do not have regular internet access on this computer.



This window facilitates the first and last step of the offline activation process, which is described on the window itself. To begin the process, click *Save registration request to file*. In the file chooser window that appears, select the folder where you would like this file saved.

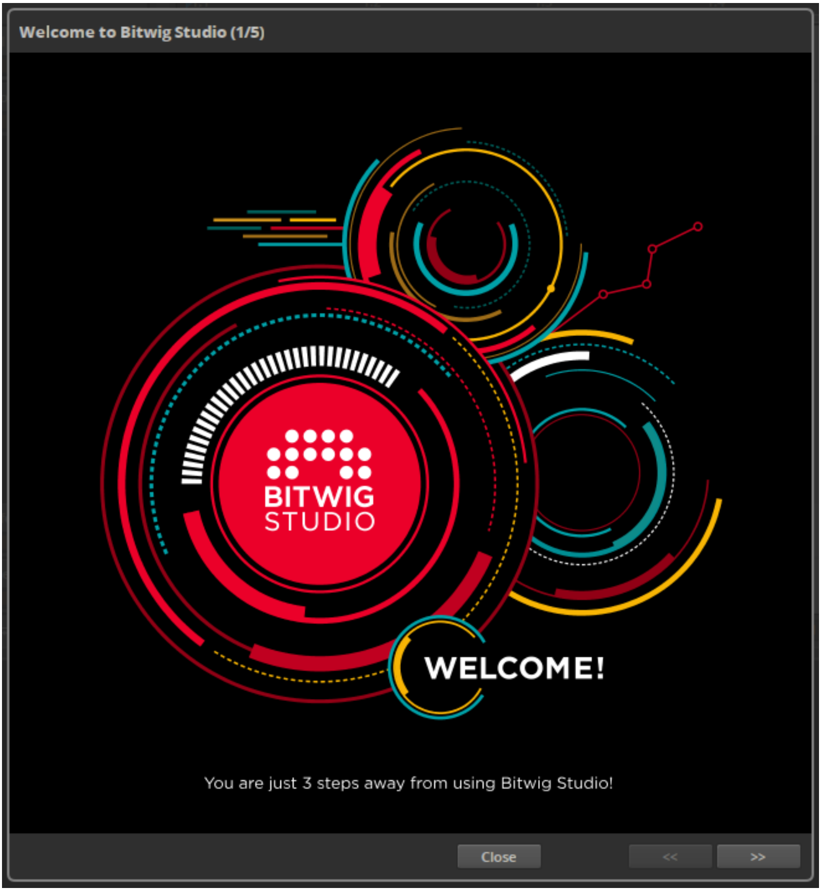
Once you have uploaded that file to Bitwig's website and downloaded the response file, click *Choose registration response file*. In the file chooser window that appears, select the response file.

› *Run in demo mode* is for trying out Bitwig Studio. Note that starting the program this way will disallow saving and exporting.

Once you have made it through one of the above paths, click through to continue.

0.1.2. Studio Setup Options

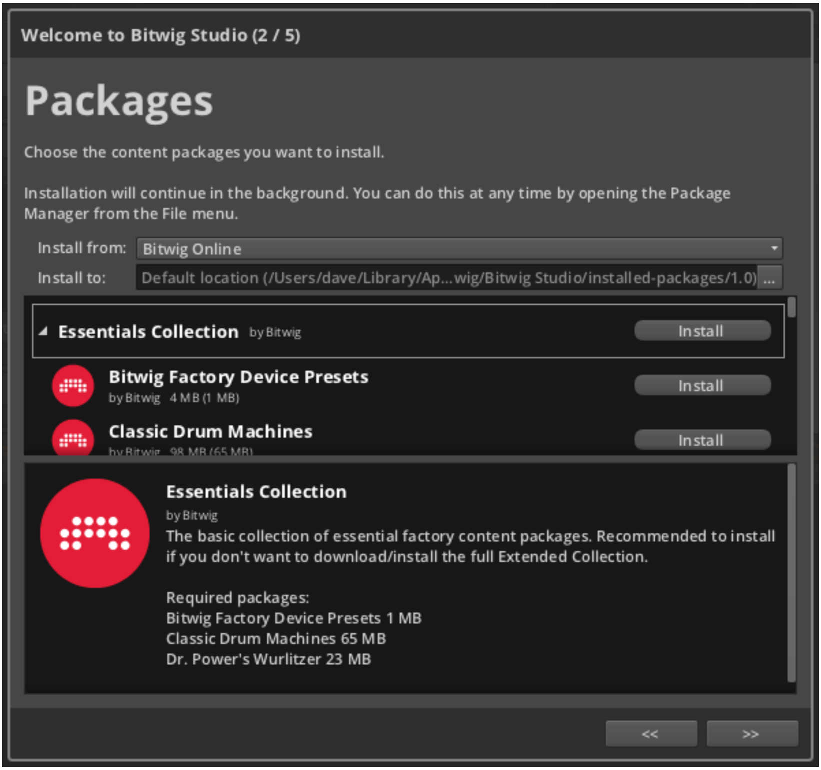
The next set of windows will help you configure three aspects of Bitwig Studio. The first in the sequence is a welcome window.



While you are free to click the *Close* button and end this short configuration sequence, we recommend that you click the next button (>>) to continue to the next page. Either way, you will be able to revisit these options and settings later.

0.1.2.1. Installing Content Packages

The next window allows you to install content packages for Bitwig Studio. These packages supply presets, sample instruments, and other media for using in your Bitwig Studio projects.

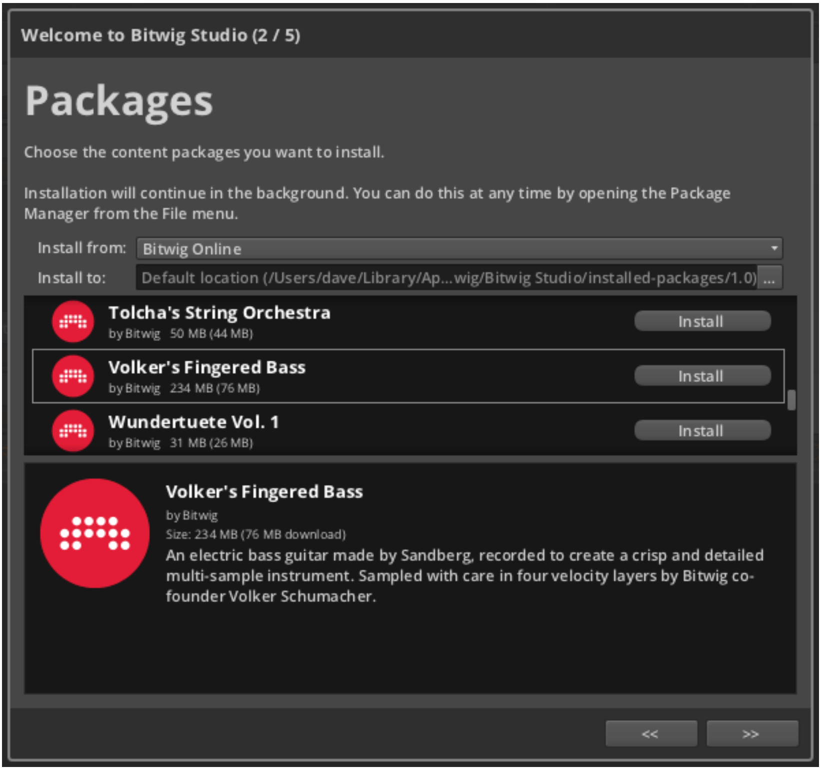


The default behavior of this window is to download the content from Bitwig's servers. This requires an internet connection.

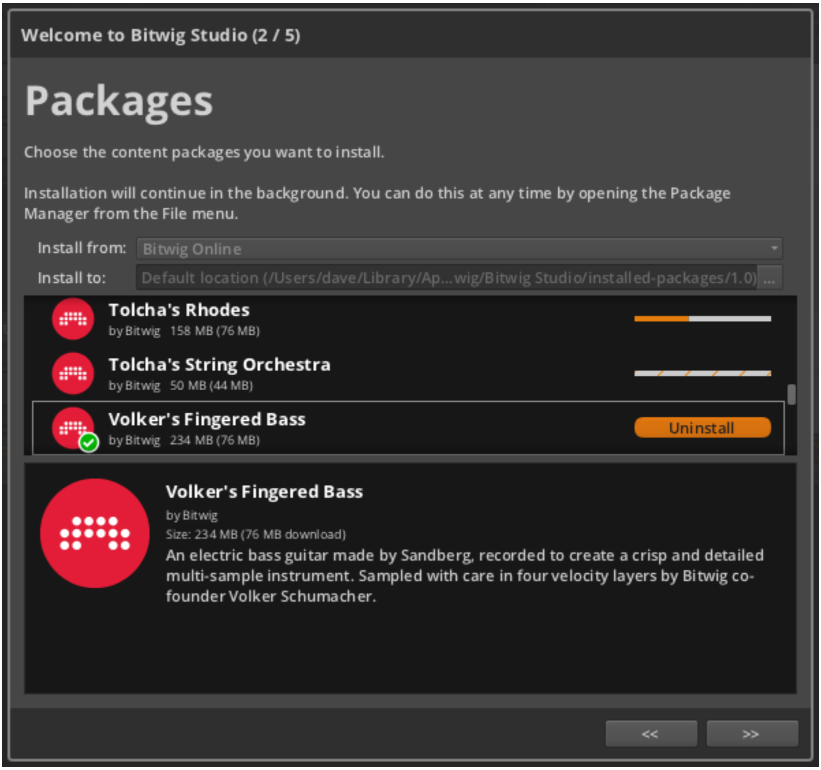
If you have a boxed copy of Bitwig Studio and wish to install this content from the included DVD, switch the *Install from:* menu selection to *Choose location...*. In the file chooser window that appears, select the Bitwig Studio DVD.

Also by default, the installer places all installed content and other preferences within your current user's folder. By clicking the ellipsis icon (...), a file chooser will appear that allows you to select another location for installation. (Changing this setting will also move any preexisting library file to the new location.)

Below the *Install to:* menu is information about the various available content packages. If you scroll through the list, you will find each package's name along with the amount of disk space it requires and the size of the download (in parentheses). By clicking on a package's name, a description of that package will be shown.



You can install packages individually by clicking their own *Install* buttons at the right end of their entries, or you can install entire categories of packages (such as *Essentials Collection*, *Extended Collection*, etc.) simultaneously by clicking their *Install* buttons. Once an installation has begun, the *Install* buttons become progress bars.



In the example above, the first package is in the process of downloading/installing, the second is waiting for its turn, and the third has finished (note the check mark on the package icon).

Once you have started installing the content packages of interest, you have done your part and this process will continue in the background. You can click the button (>>) to proceed.

0.1.2.2. Setting Up an Audio Interface

The next window allows you to designate the audio interface that you will be using with Bitwig Studio. This could be your computer's built-in audio interface or an external device that has any necessary drivers installed already. At first, this window is just a series of menus.

Begin by selecting the proper *Audio System* for your interface. The options available here vary based on your platform. If you are unsure of what to set, try the first option available (there may be only one option).

The *Input Device* and *Output Device* settings specify which audio interface you will be using for bringing audio signals into and out of the system, respectively. Whether you plan on using audio input or not, you must set the *Output Device* in order to hear anything out of Bitwig Studio.

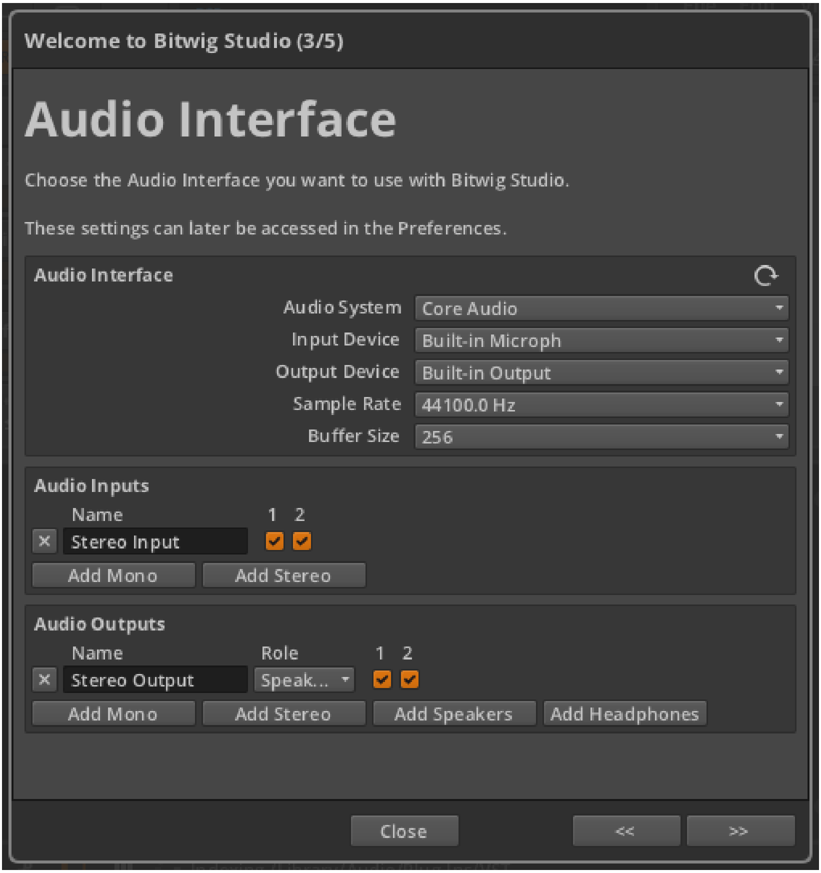
Once the *Output Device* is selected, the *Sample Rate* will now have a value. Also, the *Audio Outputs* section of the window will have contents. Bitwig Studio will have created a stereo output pair that is mapped to the first two audio outputs of your interface.

In the example shown above, the stereo output created by Bitwig Studio was named *Stereo Output*. The *Output Device* selected (*Built-in Output*) has only two available audio outputs, which is indicated by the boxes labeled *1* and *2*. The fact that both boxes are checked means that they are being used for the *Stereo Output* path, which will be available in the program under that name.

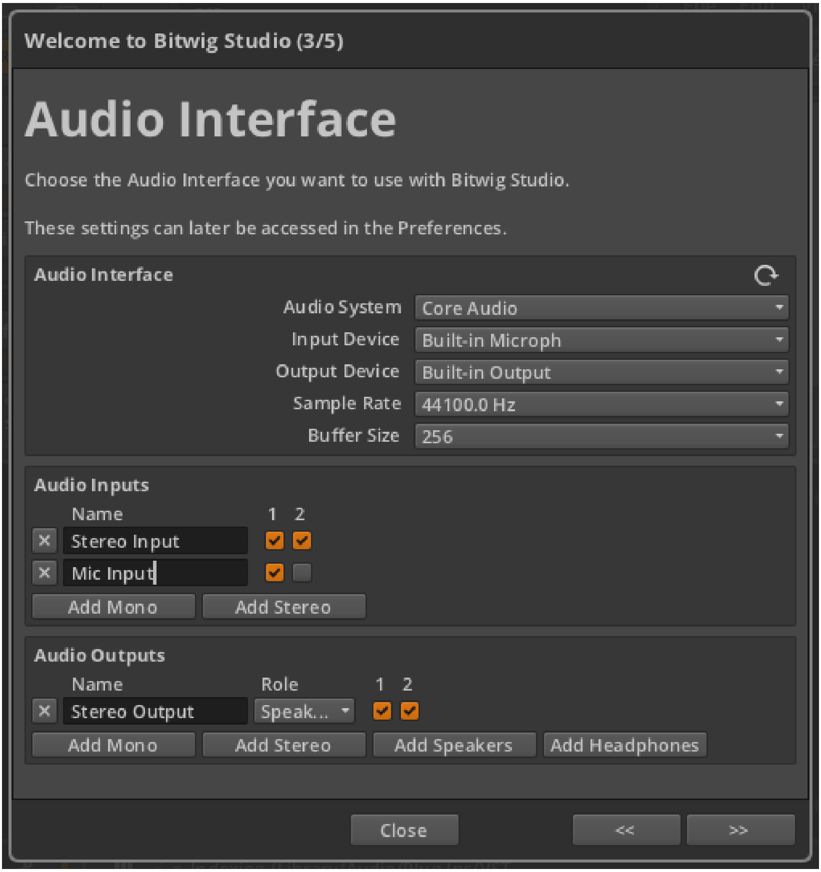
Finally, each output path has an assignable *Role*. The *Stereo Output* path has been defined as *Speakers*, making it an option for audio monitoring.

The other *Role* settings are *Headphones* (also a monitoring option) and *Output*, which covers anything other than speakers or headphones.

If we select an *Input Device*, a *Stereo Input* will be similarly created from the first two inputs.



As an example, let's say that I sometimes want to record just the first channel of the *Built-in Microph*(one) interface. By clicking the *Add Mono* button in the *Audio Inputs* section of the window, a new input path will be created with only one audio input enabled. And we can rename this path anything we like by selecting the *Name* parameter and using the computer keyboard.



Finally, the *x* button at the beginning of each line will delete that path. So if you create a path by mistake, just click this button.

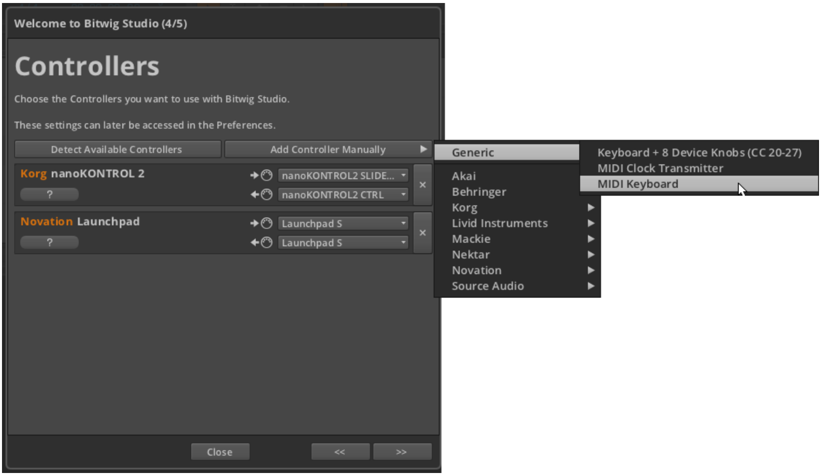
Once you are happy with the settings here, you can click the button (>>) to proceed.

0.1.2.3. Setting Up MIDI Controllers

The next window allows you to designate any MIDI controllers that you will be using with Bitwig Studio.

If you have MIDI controllers connected to the computer, you should start by clicking the *Detect Available Controllers* button, which will then find any known controllers and add them for your use.

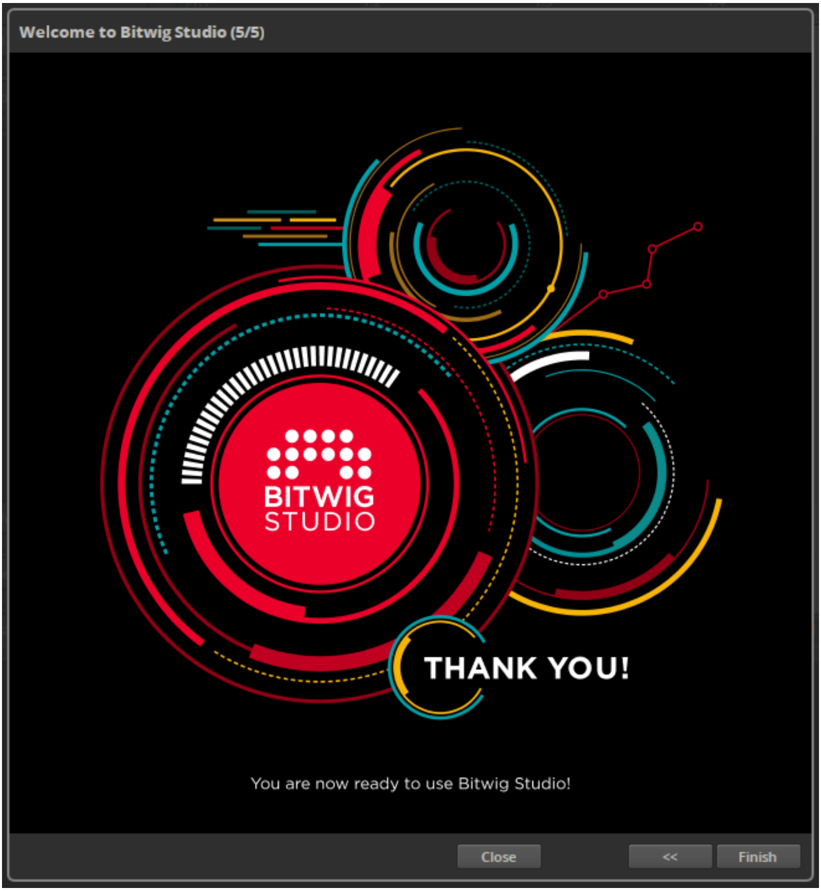
Any controllers that are missed by this scan can be added manually by clicking the *Add Controller Manually* button. A menu will appear listing various manufacturers, each containing a submenu of models. If you do not find your device here, you can choose the top menu item, labeled *Generic*.



In my case, I am using a small MIDI keyboard with no additional controllers so I will choose *MIDI Keyboard*. (If you have a keyboard with any knobs or faders, try using *Keyboard + 8 Device Knobs (CC 20-27)*.)

All that is left is to select the MIDI input port that will be used by this device by clicking on the menu that is currently set to *None*.

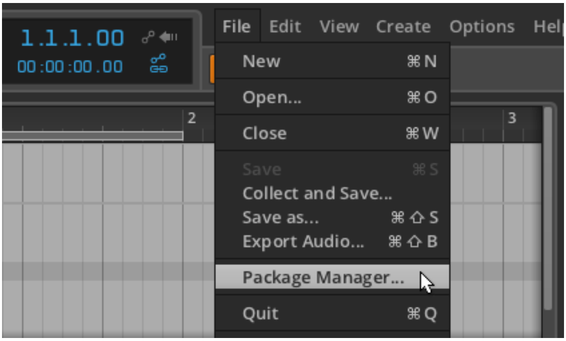
Once you are happy with the settings here, you can click the button (>>) to proceed. And the next window will explain itself.



0.1.3. Finding the Studio Setup Options Later

After you're set up and using Bitwig Studio, all of these settings can be changed at any time.

Content Packages can be installed and managed through the **Package Manager**, which is available under *File › Package Manager...*.



The audio interface and MIDI controller settings can be accessed in the Preferences window, which is available under *Options* › *Preferences*. The settings will be found respectively in the *Audio* and *Controllers* tabs.

0.2. Document Conventions

Here are a few notes on the formatting of this document, particularly in relation to the platform you may be using:

› Whenever key commands are the same for Windows, OS X, and Linux, the command will be listed once without any comment. When the key command is different for the platforms, the Windows/Linux version will be listed first, and the Mac version will follow and be labeled. An example for the copy function would be: press [CTRL]+[C] ([CMD]+[C] on Mac).

› If you are on a Mac, your [ALT] key might be labeled "option." In this document, it will always be called [ALT].

› If you are on a Mac, your "command" key might be labeled with an apple icon. In this document, it will always be called [CMD].

› If you are on a Mac, right-clicking can also be achieved by [CTRL]-clicking.

› Screenshots in this document were made with the Mac version of Bitwig Studio.

1. Bitwig Studio Concepts

This chapter is both an introduction to the program and an overview of its structure. Please start here to get acquainted with the fundamental concepts and related vocabulary used in Bitwig Studio.

1.1. Top-Level Concepts

Bitwig Studio is a modern digital audio workstation (DAW) that allows you to seamlessly compose, produce, perform, and expand your music.

Any session that you create in Bitwig Studio is called a *project*. You can have multiple projects open at once, but audio will be active for only one of these projects at a time.

Bitwig Studio projects are organized into *tracks*, which can be thought of as either individual instruments or layers that should be handled similarly. Each track contains a signal path that results in audio and has common mixing board controls (such as volume, panning, solo, and mute).

Clips are containers for individual musical ideas. Clips store either notes or audio, as well as control and automation data.

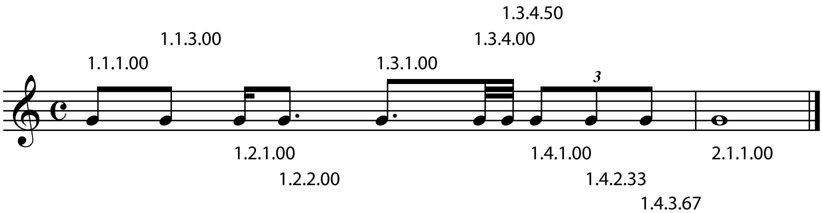
Music is made in Bitwig Studio by creating a project and populating its tracks with clips, which you can then refine, arrange, and trigger.

1.2. A Matter of Timing

As Bitwig Studio's primary task is to record and play back music, the element of time is crucial. The *transport* (most closely associated with the global play, stop, and record buttons) is the engine that drives all time functions in Bitwig Studio. This means that for any clip(s) to be played back, triggered, or recorded, the transport must be active, propelling the Global Playhead forward.

Bitwig Studio works with time in musical units of bars, beats, and ticks (a set subdivision, which defaults to sixteenth notes). A final value is stored for finer resolution, which is a rounded percentage of the distance between the current tick and the next one. These four units are shown together with period spacers in this way: `BARs.BEATs.TICKs.%`

For example, with a default time signature setting of *4/4*, *1.3.4.50* would represent an event happening in the first bar, on the third beat, within the fourth sixteenth note, exactly halfway to the next sixteenth note. The example below uses Bitwig Studio's counting system to label a rhythm in traditional musical notation:



1.3. One DAW, Two Sequencers

Within Bitwig Studio are two independent sequencers:

› The *Arranger Timeline* (or *Arranger*) is a linear sequencer that operates across a standard musical timeline. This is the place for sketching and producing full-length songs or other works.

› The *Clip Launcher* (or *Launcher*) is a nonlinear sequencer where you can accumulate a bank of musical ideas and then mix and match them. Clips in the Launcher can be organized into groups called *scenes*, either for triggering those clips together or for composing in blocks (such as verse, chorus, bridge, etc.).

The Arranger Timeline and Clip Launcher contain completely separate data. Editing clips on the Arranger Timeline has no effect on those stored in the Clip Launcher, and vice versa. But the Arranger Timeline and Clip Launcher do interact in several critical ways:

› Clips can be freely copied between the Arranger Timeline and Clip Launcher. When selected together, multiple clips can also be copied back and forth, and scenes can as well.

› The result of all triggered Launcher clips can be recorded directly to each Arranger track, allowing you to capture an improvisation that can be edited later.

› Except when recording the Clip Launcher's output to the Arranger Timeline, only one of these two sequencers is active at any given time. So on a track-by-track basis, you choose whether the Arranger Timeline or Clip Launcher is in control and can trigger its data.

› By default, the Arranger Timeline is the active sequencer for each track.

› Each track can play only one clip at a time.

1.4. Devices and Other Signal Achievements

Devices are special-function components that extend your signal paths by modifying or transforming incoming notes or audio signals.

Every track has a *device chain*. In terms of signal flow, this device chain falls between the incoming sequencer data and the track's mixing board section. In this device chain you can insert as many devices as you like. You can even use Bitwig's devices to create additional device chains.

Each device has *parameters*, which are settings that determine how that device operates. Parameters are set directly within the device's interface or via an assigned MIDI controller. Parameter values can also be sequenced via automation or adjusted along with other parameters via macros.

Devices are grouped into the following functional categories:

› *Note FX* manipulate incoming note messages before passing them onward.

› *Instruments* are devices that use incoming note messages to synthesize audio.

› *Containers* are utility devices whose primarily function is to host other devices.

› *Audio FX* manipulate incoming audio signals before passing them onward.

› *Generators* are devices that produce output without the need for any input.

› *Modulators* are devices that can modulate parameters within their own *FX* chain. Modulators are primarily used to manipulate other devices, but they can also influence some of their own parameters.

› *Routers* divert a track's signal path, allowing signals to exit and/or reenter the track. This includes the ability to route signals outside of Bitwig Studio itself (for access to hardware synthesizers and effects units, etc.).

All device chains in Bitwig Studio support both audio and note signals. To keep these signals accessible, a few rules apply.

› Except for note FX devices, all devices receiving note signals pass them directly to their output. (Note FX process the incoming notes before passing them onward.)

› Except for audio FX devices, all devices receiving audio signals pass them to their output. (Audio FX process the incoming audio before passing them onward.)

› Many Bitwig devices possess a *Mix* parameter. Similar to a "wet/dry" fader, this control blends the raw audio that entered the device into the device's output.

In Bitwig Studio, all audio signal paths are stereo.

1.5. A Musical Swiss Army Knife

Bitwig Studio's various viewers and editors are called *panels*. These panels are the heart of the program and the places where all work happens.

› The **Arranger Timeline Panel** lets you see all of your project's tracks, create an arrangement with timeline clips, and edit track automation.

› The **Clip Launcher Panel** allows you to trigger clips both freely and in sync with the transport, copy clips into and out of the arranger, and sort clips into scenes.

› The **Detail Editor Panel** is the graphical editor for both notes and audio, and their affiliated data.

› The **Inspector Panel** displays all parameters for any selected clips, notes, audio events, or tracks.

› The **Mixer Panel** presents the channel strip for each track and any subsidiary signal chains.

› The **Device Panel** shows the full device chain for the selected track, including an interface for each Bitwig device and VST plug-in in use.

› The **Automation Editor Panel** gives you detailed control over track automation, clip automation, and MIDI control messages.

› The **Browser Panel** allows you to preview, load, save, and tag content from your Bitwig Studio library and elsewhere on your machine.

› The **Project Panel** manages your project's metadata as well as the status of files and plug-ins being used.

› The **Studio I/O Panel** gives assorted audio and MIDI options, such as routing the main audio buss to any pairs of speakers and headphones, listing the current MIDI controller mappings, etc.

The primary interfaces in Bitwig Studio are called *views*. Each view gives you access to a set of panels chosen to help you carry out a particular musical job.

› The **Arrange View** lets you focus on assembling music, particularly by recording and ordering clips. The **Arranger Timeline Panel** is central to this view along with the optional **Clip Launcher Panel**. All panels are available here, and all project tracks are viewed together.

› The **Mix View** focuses on mixing tracks and triggering clips. The **Mixer Panel** is central to this view along with the optional **Clip Launcher Panel**. Except for the **Arranger Timeline Panel**, all other panels are available here, and all project tracks are viewed together.

› The **Edit View** is for making detail edits to clips. The **Detail Editor Panel** is central to this view along with the optional **Automation Editor Panel**. Except for the **Arranger Timeline**, **Clip Launcher**, and **Mixer** panels, all other panels are available here.

Bitwig Studio offers several window arrangements called *display profiles*. These configurations adjust the placement of panels and even provide additional application windows when appropriate. This is all in the name of optimized workflows, allowing the program's layout to match your current screen arrangement and the task at hand.

› *Single Display (Large)* is intended for use with one monitor, using a single application window to focus on one of Bitwig Studio's views at a time. *This is the default display profile (and the one used for screenshots within this document).*

› *Single Display (Small)* is similar to the *Single Display (Large)* profile but is optimized for use on a smaller monitor.

› *Dual Display (Studio)* is intended for use with a two-monitor setup, such as a laptop screen and an external display. This profile keeps the **Arrange View** on your primary display and toggles your secondary display between the **Mix View** and the **Edit View**.

› *Dual Display (Arranger/Mixer)* is intended for use with a two-monitor setup. This profile is fixed, keeping the **Arrange View** on your primary display and the **Mix View** on your secondary display.

› *Dual Display (Master/Detail)* is intended for use with a two-monitor setup. This profile keeps the **Edit View** on your secondary display and toggles your primary screen between the **Arrange View** and **Mix View**.

› *Triple Display* is intended for use with a three-monitor setup. This profile is fixed, keeping the **Arrange View** on your primary display and the **Mix View** and **Edit View** on your secondary and tertiary displays.

1.6. User Interfacing

Finally, a few notes to help you interact with Bitwig Studio.

› Any interface control (like a knob or curve control) can be set with the mouse by clicking and dragging upward or downward. You can [CTRL]-click ([CMD]-click on Mac) on the control to set its value with the keyboard. Double-clicking on the control restores its default value.

› Any numeric control (one that directly shows you numbers) can be set with the mouse by clicking and dragging upward or downward. You can also double-click on the control to set its value with the keyboard.

› Any control at all can be fine-tuned with the mouse by [SHIFT]-clicking the control and dragging. If you have already clicked the control, you can also press [SHIFT] after the fact to engage this mode.

› When a button is tinted orange, that control is active. The inactive form of a control uses a neutral color, such as white, gray, or silver.

› Many key commands remain available while you are clicking and dragging an item. These include the commands for toggling panel visibility or switching the current view.

› Only one visible panel will ever have focus at a given time. Focus follows the panel that was last clicked or activated. Panel focus is indicated by the outer rounded rectangle being tinted silver. Key commands that target a specific panel are available only when that panel is in focus.

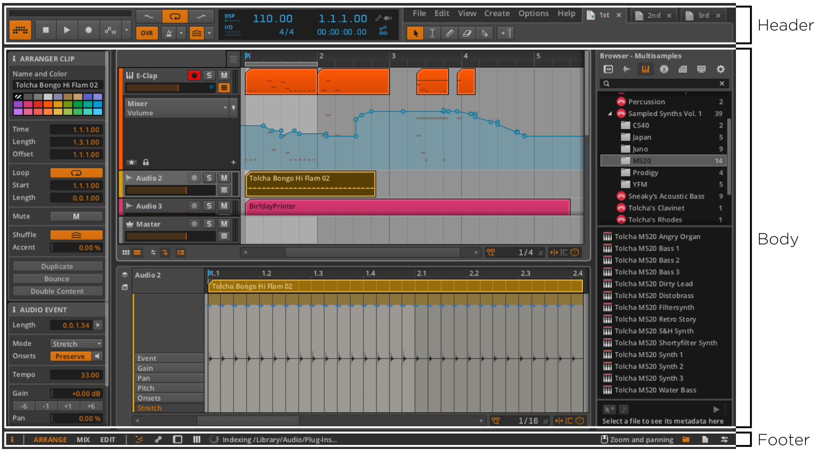
› Enabling [CAPS LOCK] causes your computer keyboard to transmit note messages. While this can be a quick way to enter notes, it will also disable many normal key commands. If your key commands are not working, make sure that [CAPS LOCK] is disengaged.

› The **Commander window** allows you to find and trigger functions from across Bitwig Studio.

  To call up the Commander window: select *Help* › *Commander...* or press [CTRL]+[ENTER]. You can then scroll to browse all program functions, or type to search them.

2. Anatomy of the Bitwig Studio Window

All functions and controls of Bitwig Studio are accessible through the application window. Each window can be thought of in three vertical slices: the header, the body, and the footer.



We will first examine the reliable header, then the pliant footer, and finally the mercurial body.

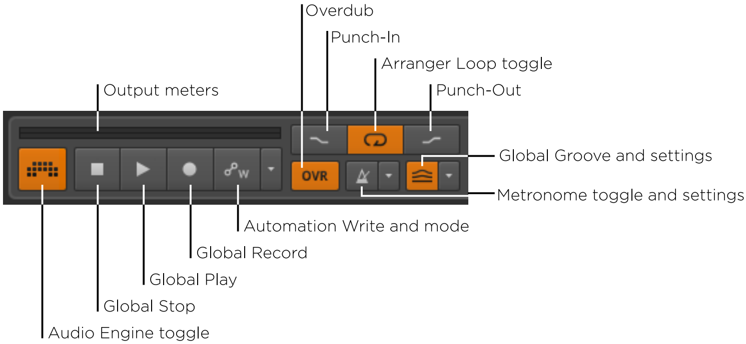
2.1. The Window Header

The header of each window is divided into three horizontal sections: transport controls, display, and the toolbar.

These sections and their fixed controls are always present.

2.1.1. Transport Controls Section

The window header's transport controls section provides various buttons and controls for manipulating the transport's behavior. (Bitwig Studio's transport drives the Global Playhead and thereby manages all time-based events, including playback and recording.)



This section contains the following items:

› *Output meters*: Stereo audio meters that mirror those of the master track, displaying the level reaching the main audio output buss.

› *Audio Engine toggle*: Enables/disables the audio engine for the current document. While disabled, most of the other controls in this section will be unavailable. (The audio engine can be engaged for only one Bitwig Studio project at a time, no matter how many are open.)

› *Global Play*: Toggles and indicates the state of Bitwig Studio's transport. When clicked to toggle the transport on, Arranger playback resumes from the Play Start Position and active Launcher clips are triggered in sync. When clicked to toggle off, the transport is stopped and the Play Start Position is moved to the current Global Playhead position.

› *Global Stop*: Deactivates the transport. When the transport is already inactive, clicking the global stop button returns both the Global Transport and the Play Start Position to the beginning (play position *1.1.1.00*).

› *Global Record*: Arms all record-enabled tracks. When the global record button is enabled, Arranger recording will begin the next time the transport is started.

› *Automation Write and mode*: Enables automation recording to the Arranger Timeline the next time the transport is started. The drop-down menu on the right provides automation recording modes and *Pre-roll* options (1, 2, or 4 bars, with or without metronome).

› *Punch-In*: Causes recording to begin at the start of the Arranger Loop Selector.

› *Arranger Loop toggle*: Activates/deactivates Arranger looping within the Loop Selector's bounds.

› *Punch-Out*: Causes recording to stop at the end of the Arranger Loop Selector.

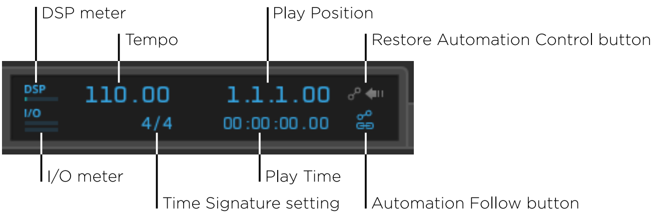
› *Overdub*: Preserves notes already present when Arranger recording takes place. Otherwise, note data is overwritten.

› *Metronome toggle and settings*: Enables/disables the metronome whenever the transport is active. The drop-down menu on the right provides settings for volume (*Vol.*) and whether sub-beats (*Ticks*) should also sound.

› *Global Groove and settings*: Enables shuffle for all clips whose own *Shuffle* parameter is enabled. The drop-down menu on the right provides settings for the *Shuffle* interval and *Amount*, as well as the *Accent* interval, *Amount*, and *Phase*.

2.1.2. Display Section

The window header's display section provides informational meters, numeric controls, and a couple of automation-related settings.



This section contains the following items:

› *DSP meter*: Displays Bitwig Studio's current CPU usage.

› *I/O meter*: Displays Bitwig Studio's current disk activity for data being read (input) and written (output), respectively.

› *Tempo*: A control for the project's current tempo, set in beats per minute (BPM).

› *Time Signature* setting: A control for the project's current time signature and an optional tick setting.

The time signature itself accepts common denominators (such as *2*, *4*, *8*, and *16*) that represent the type of beats counted in each bar (half, quarter, eighth, and sixteenth notes, respectively).

The time signature's numerator represents the number of beats in each bar. It can be set as a positive integer (such as *7*) or as the sum of several integers (such as *3+2+2*). Both versions (*7/4* or *3+2+2/4*) would be treated the same by Bitwig Studio.

The optional *tick* setting represents the primary beat subdivision to be used across the project (see section 1.2). If only a time signature is set (like *4/4*), a default tick setting of sixteenth notes is used. If the time signature is followed by a comma and an appropriate tick value (such as *4/4,8*), then that tick setting will be used. Values recognized by Bitwig Studio include *8* (eighth notes), *12* (triplet eighth notes), *16* (sixteenth notes), *24* (triplet sixteenth notes), *32* (thirty-second notes), and *48* (triplet thirty-second notes).

› *Play Position*: A control for the project's current play position, shown as `BARs.BEATs.TICKs.%`.

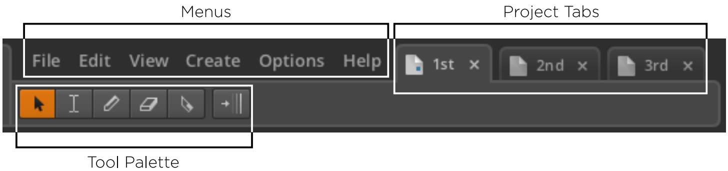
› *Play Time*: A control for the project's current play time, shown as `HOURs:MINUTEs:SECONDs.MILLISECONDs`.

› *Restore Automation Control button*: Restores control of automation after a parameter is adjusted during playback. The Restore Automation Control button arms itself when the function is useful.

› *Automation Follow button*: Toggles whether automation is moved along with clips or not.

2.1.3. Toolbar Section

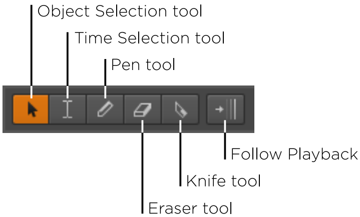
The window header's toolbar section consists of Bitwig Studio's tool palette, menus, and project tabs.



While the tool palette and menus are consistent, the project tabs shown will vary, matching what projects you have open at a given time.

2.1.3.1. Tool Palette Section

When working in Bitwig Studio, one of the five tools will be active at any given time. The selection of tool determines how the mouse functions within the program's timeline-based panels (the **Arranger Timeline Panel**, the **Detail Editor Panel**, and the **Automation Editor Panel**).



This section contains the following items:

› *Object Selection tool* is for selecting and moving events. Clicking between automation points along the current curve will create a new point. And double-clicking in a blank area will create a new event of the appropriate kind. You can switch to this tool by pressing [1], or you can temporarily use the tool by holding [1].

> **Note**
>
> As this is the primary tool in Bitwig Studio, all editing functions described in this document presume you have the Object Selection tool engaged. If a different tool is meant to be used, it will be specifically noted.

› *Time Selection tool* is for choosing an arbitrary section of time instead of particular events. Otherwise it generally acts like the Object Selection tool. You can switch to this tool by pressing [2], or you can temporarily use the tool by holding [2].

› *Pen tool* is for drawing new events. You can switch to this tool by pressing [3], or you can temporarily use the tool by holding [3].

› *Eraser tool* is for deleting relevant events from the area of time that you select. You can switch to this tool by pressing [4], or you can temporarily use the tool by holding [4].

› *Knife tool* is for splitting a continuous event into two. You can switch to this tool by pressing [5], or you can temporarily use the tool by holding [5].

› The *Follow Playback* option causes all timeline-based panels to keep the Global Playhead on screen at all times.

> **Note**
>
> In the *General* tab of the Preferences window are two settings for the *Playhead follow mode*:

› *Scroll by pages* will scroll once the Global Playhead reaches the edge of the current display area. This is the default setting.

› *Continuously scroll* will keep the Global Playhead centered in each timeline-based panel.

2.1.3.2. Menu Section

The menus themselves are fairly standard, containing relevant items based on the menu names.

This section contains the following menus:

› *File* lets you create, open, and save Bitwig Studio projects in various ways. It also lets you install additional program content or quit the application.

› *Edit* lets you apply different functions to the current selection, as well as undo (or redo) recent actions across the program.

› *View* lets you choose which interface items to display, change the display profile being used, and view the Control Script Console.

› *Create* lets you make new tracks, scenes, and events.

› *Options* lets you adjust the way the program operates and provides access to Bitwig Studio's Preferences window.

› *Help* lets you view the program's documentation, access the **Commander window**, and check with Bitwig for software updates.

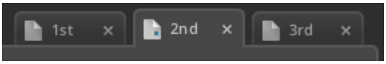
2.1.3.3. Project Tab Section

On the far right are tabs for the Bitwig Studio projects which are currently open. Below are some notes on using these tabs:

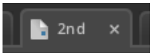
› Bitwig Studio will display the contents of only one project at a time. This is true even if you are using a display profile that uses multiple application windows.

› To focus on any one of the open projects, click on its tab.

› The tab in the foreground represents the currently viewed project. In the image below, this is the project named *2nd*.



› Only one project at a time is capable of producing sound. In the icons within the project tabs, a small blue shape in the lower right corner indicates which project (if any) has activated Bitwig Studio's audio engine. A rightward-facing triangle indicates that the transport is active, and a square indicates that it has stopped.



› You can click and drag any project tab to change its position among the project tabs.

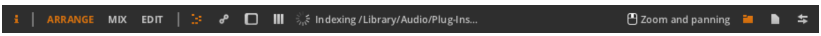
› If there is not enough space to show all open projects together, left and right scroll arrows will appear around the project tabs.

› Within each project tab, an asterisk (*) will be appended to the project's name if unsaved changes have been made.

› The *x* on the right side of each tab can be clicked to close that project.

2.2. The Window Footer

The window footer contains various buttons that determine which parts of Bitwig Studio are visible, along with various status messages and notifications.



Footers will differ based on the display profile being used. The image above — and all screenshots in this document — shows a footer from the default *Single Display (Large)* profile in **Arrange View**, where all panels and views are available.

2.2.1. Panel Icons

The small icons that appear in the window footer are panel icons. Each icon represents a panel that is available within the current view. The icons are also buttons, allowing you to toggle the visibility of each panel by clicking its icon. An icon that is illuminated in orange indicates an active panel.

For each cluster of icons, only one panel can be shown at a time. These icon clusters are located either on the far-left, far-right, or center-left of the window footer, indicating whether those panels would be displayed on the left, right, or center-bottom of the window, respectively.

The panel icons that you will encounter are:

- *i* The **Inspector Panel** icon is a seriffed, lowercase *i*. When available, you can focus on this panel and toggle its visibility by pressing [I] or [ALT]+[I].
- The **Detail Editor Panel** icon is an arrangement of dashed lines, like a standard "piano roll" representation of notes. When available, you can focus on this panel and toggle its visibility by pressing [E] or [ALT]+[E].
- The **Automation Editor Panel** icon is two circles connected by a line, like the breakpoints that build an automation curve. When available, you can focus on this panel and toggle its visibility by pressing [A] or [ALT]+[A].
- The **Device Panel** icon is a rounded rectangle with a shaded left side, like the containing box for each device and its left-sided title bar and master controls. When available, you can focus on this panel and toggle its visibility by pressing [D] or [ALT]+[D].
- The **Mixer Panel** icon is a series of three wide vertical lines, like the volume faders of a mixing console. When available, you can focus on this panel and toggle its visibility by pressing [M] or [ALT]+[M].
- The **Browser Panel** icon is a folder icon, representing the library of content that is accessible in this panel. When available, you can focus on this panel and toggle its visibility by pressing [ALT]+[B].
- The **Project Panel** icon is a file icon, representing the project file whose metadata is defined in this panel.
- The **Studio I/O Panel** icon is a pair of opposite-pointing arrows, representing the input and output paths that are addressed in this panel.

2.2.2. View Words

The capitalized, bold words that appear on the left side of the window footer represent all currently available views. To match the views' names, the labels used are *ARRANGE*, *MIX*, and *EDIT*.

A window with no view words indicates that your current display profile is fixed and has only one available view.

For the two-window display profiles (those whose name begins with *Dual Display*), available views are shown as compound names, such as *ARRANGE-MIX* or *MIX-EDIT*. In this situation both windows show the same view words, indicating the views shown on the primary and secondary windows, respectively.

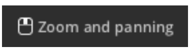
2.2.3. Status Messages and Notifications

Status messages appear just to the right of all left-aligned view words and panel icons.

One function of status messages is to list additional options for the task you are doing. So once you click and begin to drag an object, the status bar will list all modifier keys that can tweak your action. A common example is noting when [ALT] transforms a move function into copy, and vice versa.

Another function is to indicate that a computer process is currently underway. These messages are often accompanied by a spinning gear icon to suggest activity, and a progress bar when the process's status can be assessed. Once the status message goes away and nothing replaces it, the process has completed. Common examples are the scanning of new VST plug-in folders (without showing a progress bar) or an audio bounce function (with a progress bar).

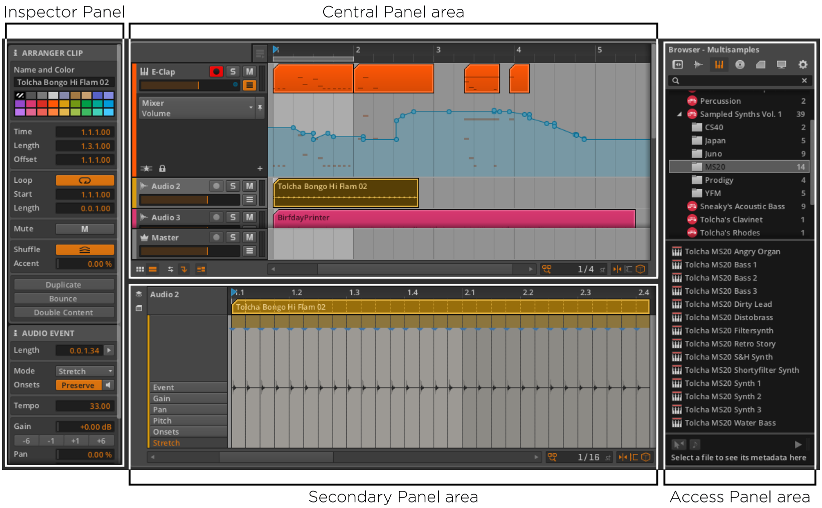
Notifications appear on the right side of the window footer, just before any right-aligned panel icons. They function more as tooltips, letting you know a function is available that might be missed. A common example is when a middle mouse button (should you have one) directly triggers a different function.



2.3. The Window Body

So the window header is always the same (aside from the project tabs), and while the footer's content and arrangement depend upon the current display profile, the set of controls is consistent. These two areas give you control of the program and its behavior so they are generally static. Not so with the window body.

The window body's purpose is to display your work so that you can edit it in different situations. To that end, the body's appearance is always changing, giving you the tools you need to perform specific tasks, but certain areas of the window body are designated for consistent usage.



The central portion of the Bitwig Studio window is reserved for the *central panel*. The panel(s) shown here is defined by the window's current view (either **Arrange**, **Mix** or **Edit View**). The central panel cannot be hidden, so if all other panels were disabled, the central panel would take up the entire window body.

Below the central panel is the *secondary panel area*. This area is where a second panel can be loaded for editing your project's content. Again, the selection of available panels is determined by the window's current view and the display profile being used. Most secondary panels can be vertically resized.

On the right side of the window body is an *access panel area*. This area is usually reserved for panels that deal with things other than the content of your project. Typical access panels are the **Browser Panel** (which gives access to the Bitwig Studio library and outside files), the **Project Panel** (which gives access to the project's metadata and dependencies), and the **Studio I/O Panel** (which gives access to your hardware routings). Each of these panels can be horizontally resized. When no panel is loaded in this area, the central and secondary panels simply reclaim the space.

On the left side of the window body is an area usually reserved for the **Inspector Panel**. In certain display profiles, however, the **Inspector Panel** is included in the access panel area. This panel is not resizable.

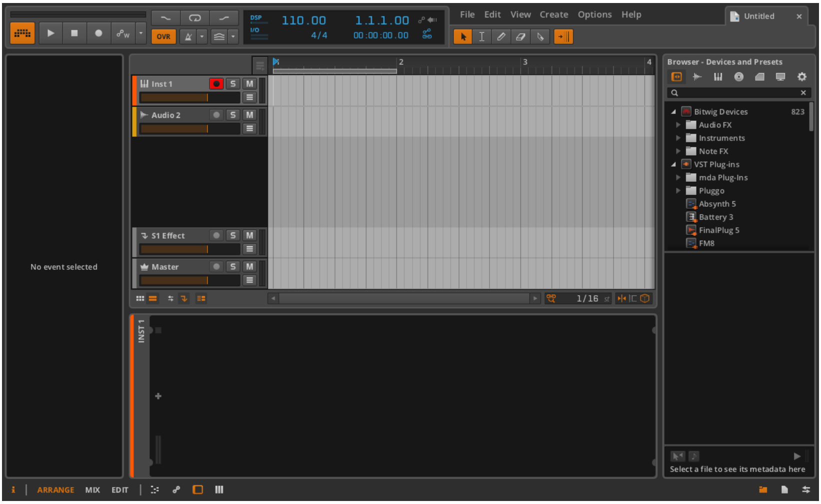
3. The Arrange View and Tracks

Now that we have examined all the fixed parts and dynamic possibilities of the Bitwig Studio window, let's enter the practical world of the **Arrange View**. We will start by looking at a few key sections of the **Arranger Timeline Panel** and their constituent elements. We will then examine the track types used by Bitwig Studio along with basic track editing functions. Finally we will get a brief introduction to the **Inspector Panel** for current and future use.

3.1. The Arranger Timeline Panel

Unlike sculpture, painting, and architecture, music is an art form appreciated over a defined length of time. That is to say, when we listen to a piece of music, either at home or out at a venue, it unfolds over the same amount of time and at the same pace for everyone in the audience. While music can definitely be performed or created with improvisation (see chapter 5), each performance has a rigidly defined structure to us listeners. And as most productions are still based around a fixed song structure, we will start with the **Arrange View** and its friend the **Arranger Timeline Panel**, which is made to lay out music arrangements in a precise way.

The **Arranger Timeline Panel** is unique in Bitwig Studio: it is available in only one view (the **Arrange View**), and it is available in this view only as the central panel. And as this panel is the only way to create a traditional, linear musical arrangement within Bitwig Studio, it is impossible to overstate the importance of the **Arranger Timeline Panel** — also called the *Arranger* — which is seen here after a new file has been created.



We will start by examining various sections of the **Arranger Timeline Panel**.

3.1.1. Arranger Area, Arranger Timeline, and Zooming

The most important element here is the actual *Arranger Timeline*, which is currently blank. As you may have seen here in earlier images (or from opening the demo session), this is the area where your song arrangements will take shape in the form of clips and track automation. Whenever we refer to an "Arranger clip," we mean a clip that is housed within this Arranger sequencer.

The Arranger is laid out horizontally, showing time progressing from the left side of the screen to the right. This can be seen in the *Beat Ruler* at the top of the Arranger. The integers here — *1*, *2*, *3*, etc. — show where each new bar begins.

To adjust the zoom level: place the mouse in-line with the bar numbers inside the Beat Ruler. The cursor will become a magnifying glass indicating that we are in *zoom mode*. Now click and hold the mouse button, dragging upward to zoom in or downward to zoom out. You can also drag the mouse from side to side to horizontally scroll within the Arranger Timeline.

Other ways to adjust the zoom level include:

› Press either [PLUS] or [CTRL]+[PLUS] ([CMD]+[PLUS] on Mac) to zoom in and either [MINUS] or [CTRL]+[MINUS] ([CMD]+[MINUS] on Mac) to zoom out.

› Hold [CTRL]+[ALT], and then click and drag anywhere within the Arranger area. If your mouse or trackpad supports a scroll function, you can also hold [CTRL]+[ALT] anywhere within the Arranger area and then scroll up and down.

› If you have a three-button mouse, click and drag the middle button anywhere within the Arranger area.

As you zoom in on the Beat Ruler, you may notice that the bar numbers start adding decimals. Depending on your zoom level, the timeline values will be represented as either `BARs`, `BARs.BEATs`, or `BARs.BEATs.TICKs`.

3.1.2. Beat Grid Settings

As you adjust the Arranger Timeline's zoom level, you may also notice that the grid lines within the Arranger area begin to change. This has to do with the *beat grid settings*, which are found in the bottom of the **Arranger Timeline Panel** and to the right of the horizontal scroll bar.

The *beat grid resolution* tells us what musical interval is being represented by the grid lines. In a new project, the *adaptive beat grid* setting is turned on (indicated by orange coloration). When adaptive beat grid is enabled, changes to the zoom level also cause appropriate changes to the beat grid resolution. The beat grid resolution setting will update as the value changes.

To toggle the adaptive beat grid: click the adaptive beat grid button within the grid settings area, or press [SLASH].

Note

On a German keyboard, the key command is [HYPHEN].

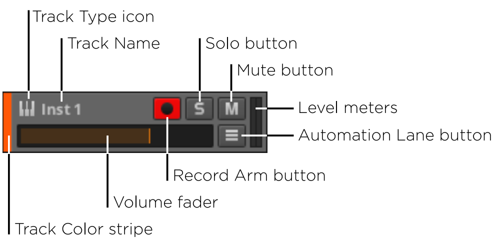
To manually set the beat grid resolution: first make sure that adaptive beat grid is disabled. Then manipulate the beat grid resolution by setting it with the mouse or by pressing [COMMA] to lower the grid resolution or [PERIOD] to raise it.

The beat grid resolution has an accompanying parameter to its right. The *beat grid subdivision* sets the rhythmic grouping used for the beat grid resolution setting. For example, the default *st* value means that straight duple values are being used. Other available settings include *3t* (triplets), *5t* (quintuplets, or fifth-lets), and *7t* (septuplets, or seventh-lets).

To manually set the beat grid subdivision: first make sure that adaptive beat grid is disabled. Then manipulate the beat grid subdivision by setting it with the mouse or by pressing [ALT]+[COMMA] to lower the grid resolution or [ALT]+[PERIOD] to raise it.

3.1.3. Track Headers

The horizontal lines you see within the Arrange area are the dividers between each track lane. To the left of the Arrange area are the *track headers*.



Within each header are the following identifications, meters, and controls for that track:

› *Track Color stripe*: A swatch of the track's assigned color.

› *Track Type icon*: An icon to indicate the kind of track.

› *Track Name*: The title assigned to the track.

› *Volume fader*: A final level control for the track.

› *Record Arm button*: Record enables the track.

› *Solo button*: When any track has its solo button enabled, only tracks with solo enabled will output their audio.

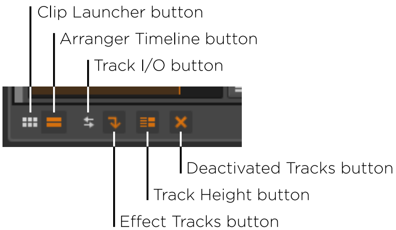
› *Mute button*: Disables the track's audio output.

› *Automation Lane button*: Toggles to reveal the automation lane section of the track (see section 8.1.1).

› *Level meters*: Stereo audio meters that display the track's output level.

3.1.4. Arranger View Toggles

Beneath the track headers are the *Arranger view toggles*. Similar to the panel icons of the window footer, each of these icons is a toggle that adjusts what is displayed in the **Arranger Timeline Panel**.



The Arranger view toggles that you will encounter are:

› *Clip Launcher button*: Toggles visibility of the **Clip Launcher Panel** (see section 5.1) within the **Arranger Timeline Panel**.

› *Arranger Timeline button*: Toggles visibility of the Arranger Timeline within the **Arranger Timeline Panel**.

> **Note**
>
> Either the **Clip Launcher Panel** or the Arranger Timeline must be visible within the **Arranger Timeline Panel**. If only one of these is visible and you hide it, the other will automatically become visible.

› *Track I/O button*: Toggles visibility of the Track I/O section of all track headers (see section 4.4.1).

› *Effect Tracks button*: Toggles visibility of effect tracks within the **Arranger Timeline Panel**.

› *Track Height button*: Toggles the track height in the Arranger between normal and half size (shown below respectively). In half size, the same track header components are displayed with some minor adjustments.

› *Deactivated Tracks button*: Toggles visibility of deactivated tracks within the **Arranger Timeline Panel**.

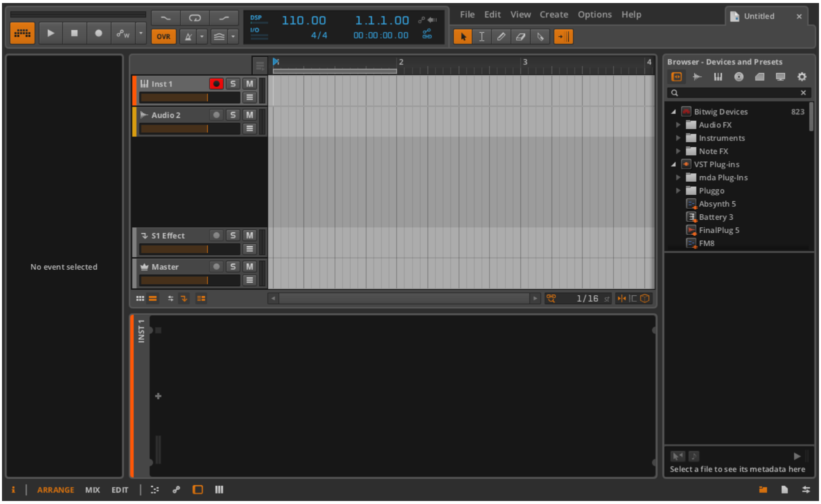
3.2. Intro to Tracks

As we have seen in the Arranger Timeline, Bitwig Studio projects are organized into *tracks*, and clips live on tracks. While clips are critical for expressing your musical ideas, tracks contain the signal paths that take clips out of the computer and into the audible world. Were there no tracks, there would be no sound either.

We will look at the kinds of tracks that exist in Bitwig Studio before discussing a few basic track operations.

3.2.1. Track Types

Bitwig Studio has five types of tracks. The four most common types are present in any new project you create. Here again is a blank new project.



As each type of track has its own designated icon, each track also has its own particular use:

An *instrument track* is denoted with a piano keys icon. The usual purpose of an instrument track is to record and hold note clips that will trigger an instrument and result in audio output.

An *audio track* is denoted with a waveform icon. The usual purpose of an audio track is to record and hold audio clips that will be played back.

A *hybrid track* is denoted with an icon that is half audio waveform and half piano keys. The usual purpose of a hybrid track is to record and hold both note and audio clips. A hybrid track is not present in a new Bitwig Studio project.

An *effect track* is denoted with a downward arrow icon. The usual purpose of an effect track is to receive portions of other tracks' audio output, then mix them together for further processing.

A *master track* is denoted with a crown icon. One and only one master track is present in each project, making him the king. The purpose of the master track is to sum all signals that are routed to the main audio buss. The master track also provides access to various transport parameters (such as tempo) for the sake of automation (see chapter 7).

3.2.2. Creating and Selecting Tracks

As you develop any project, you will almost certainly need additional tracks.

To create a track: go to the *Create* menu and select either *Create Instrument Track*, *Create Audio Track*, or *Create Effect Track*.

Other ways to create a track include:

› Use the appropriate key command as noted in the *Create* menu.

› Right-click a part of the Arranger where no tracks exist (such as the blank space between the track headers), and then choose the appropriate function from the context menu.

Before you can do anything with a track, it must first be selected, and the track header is key to this. Clicking anywhere else — including in the Arranger Timeline area — selects clips or automation, not an entire track.

When a track is not selected, the background of its header is charcoal gray, and its text and icon are light. When a track is selected, the background of its header is a light silver, and its text and icon are dark.

To select a track: click on the track's header.

When a track is already selected, you can press [UP ARROW] or [DOWN ARROW] to cycle through the adjacent tracks.

3.2.3. Edit Functions and Moving Tracks

Once a track is properly selected, several standard edit functions can be used.

To copy a track: select the track and then press [CTRL]+[C] ([CMD]+[C] on Mac).

To cut a track: select the track and then press [CTRL]+[X] ([CMD]+[X] on Mac).

To paste a track: select a track as a reference and then press [CTRL]+[V] ([CMD]+[V] on Mac). The pasted track will be added after the track that was selected.

To duplicate a track: select the track and then press [CTRL]+[D] ([CMD]+[D] on Mac).

To delete a track: select the track and then press [DELETE] or [BACKSPACE].

Other ways to execute the above functions include:

› Select the track and then choose the appropriate function from the *Edit* menu.

› Right-click the track's header and then choose the appropriate function from the context menu.

To move a track: click and drag the track's header vertically.

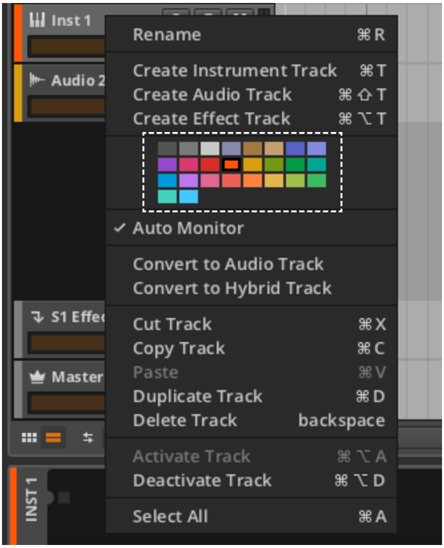
3.2.4. Track Names and Colors

You may have noticed that when a track is created, it is automatically given a name to reflect the type of track it is and its track number. And when a track is moved around, the track number in its name is dynamically updated. By default, tracks are set to automatically name themselves based on certain factors. If you desire, you can override this functionality by renaming the track.

To rename a track: right-click the track's header and then choose *Rename* from the context menu.

Each track is assigned a color when it is created. Like the track name, the track color can also be changed.

To change the color of a track: right-click the track's header and then select a different color from the palette that appears within the context menu.



3.2.5. Deactivating Tracks

There are various ways to silence a track. One useful option is to deactivate and subsequently (re)activate tracks. When a track is deactivated, not only is its output silenced, but any load it was placing on your CPU is also removed for the time being. From the standpoint of our limited computing resources, deactivating an object is as close as we can get to deleting it — and none of our data are lost in the process.

To deactivate an active track: right-click the track's header and then choose *Deactivate Track* from the context menu. Or select the track and then press [CTRL]+[ALT]+[D] ([CMD]+[ALT]+[D] on Mac).

Any disabled track that is visible has an *x* placed over its track icon and certain interface items are removed.

To activate an inactive track: right-click the track's header and then choose *Activate Track* from the context menu. Or select the track and then press [CTRL]+[ALT]+[A] ([CMD]+[ALT]+[A] on Mac).

Note

The deactivate and (re)activate functions can be applied to tracks, devices, and top-level chains/layers of the **Drum Machine**, **Instrument Layer**, and **FX Layer** container devices. And any plug-ins that are deactivated will also stop accruing latency to your project.

Similarly, clips and notes can be muted and unmuted with the same respective key commands.

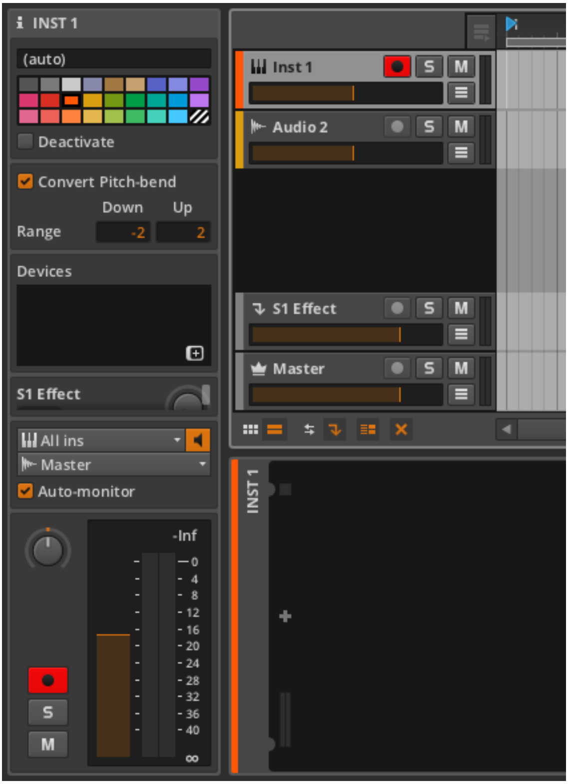
3.3. Meet Inspector Panel

A context menu is available throughout Bitwig Studio. By right-clicking on an item (practically any object or event), relevant actions that can be taken will be shown along with certain properties of that item. For a fuller list of the available properties, we also have the **Inspector Panel**.

To toggle the visibility of the Inspector Panel: select *View › Show Inspector Panel*.

The **Inspector Panel** follows the active panel's selection, displaying all properties of that selection. As there are many types of items in Bitwig Studio (clips, notes, audio events, devices, automation points, and tracks), the parameters displayed in the **Inspector Panel** can change dramatically depending on what you have clicked on.

By selecting a track, the **Inspector Panel** displays relevant parameters of that track.



We have just met some of these parameters, namely those in the *Name and Color* section at the top. The color palette is identical to the one from the track header context menu, and the *Deactivate* toggle both controls and indicates whether the selected track is deactivated or not. The text entry box above the palette is displaying the current track name. As was described in the previous section, Bitwig Studio internally names each track *(auto)* to engage the automatic track naming behavior for the track headers and other panels. Editing the track name field will override this functionality.

Plenty of other parameters are shown within the **Inspector Panel**, including nearly all of the meters and controls from the track header. And we will get to the parameters that are now unfamiliar in the appropriate sections of this document.

The main idea is that the **Inspector Panel** is an ideal way to see all the parameters of most selected items. A context menu is also available for most items and window areas. Going forward, we will primarily use the **Inspector Panel** for viewing or altering parameters and the context menu for executing functions. So this isn't "goodbye" to either option, but rather "nice to meet you."

4. Arranger Clips and the Browser Panel

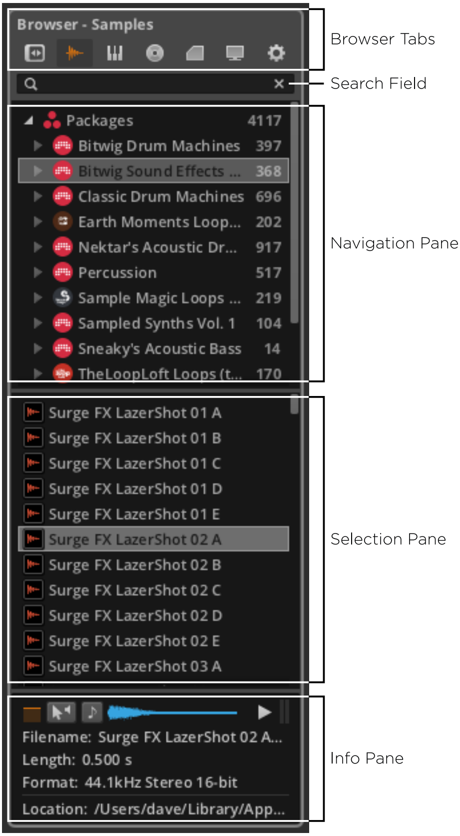
Clips are the heart of any music that you will create in Bitwig Studio. Since they are the smallest unit we will work with for arranging tasks, clips can be thought of as our musical atoms. Put a different way, a clip is the smallest musical idea that you might consider looping.

In this chapter, we will continue working with the **Arrange View**. As we want to start working with clips and adjusting their basic parameters, we must first learn about the **Browser Panel**. Then we will investigate inserting clips and moving them around in the **Arranger Timeline Panel**, which leads us to playing back Arranger contents and understanding basic transport functions. Finally, we will see how to record new clips.

If our music is made of clips, then creating and capturing our music starts here.

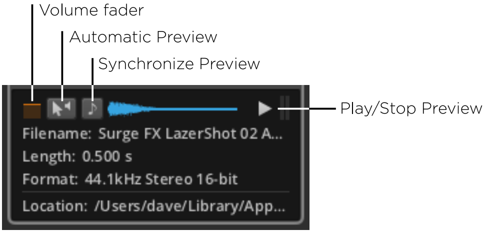
4.1. The Browser Panel

The **Browser Panel** is an organized way to access and then add various media files to your project. The files can be part of the Bitwig Studio library content or any other file found on your computer. The **Browser Panel** has a fairly consistent layout across its tabs.



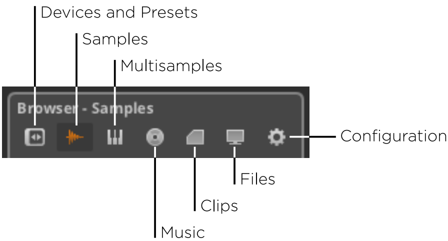
The *Browser tabs* are organized by the different types of content that are relevant to Bitwig Studio, and the search field allows you to find files/folders by name, creator, category, or tags.

The three panes below make up the file browser. The *navigation pane* at top allows you to choose from packages and folders. Once a selection is made in the navigation pane, the *selection pane* shows the contents of the selected package/folder. And once a file is picked in the selection pane, the *info pane* at bottom shows information about your file selection and offers a few options for auditioning files.



Starting on the right, the *play/stop preview* button either begins or halts playback of the selected file. When enabled, the *synchronize preview* button plays all previews at the song's current tempo. The *automatic preview* option will begin the preview of any file from the moment it is selected. And the *volume fader* at the far left determines preview volume.

To select a Browser tab, click on its associated icon.

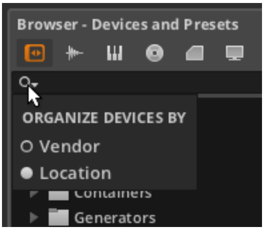


4.1.1. Devices and Presets Tab

The *Devices and Presets* tab handles Bitwig Studio's devices and their preset files as well as VST plug-ins from folders that you have designated (see section 4.1.7).

The navigation pane shows you categories of devices along with the devices and plug-ins themselves. Devices appear in this upper pane as they are considered "folders" for their presets. Those presets can be chosen from the selection pane. This is a unique arrangement since you can load items from both the navigation pane (devices and plug-ins) and the selection pane (presets) into your project.

The preview function is disabled in this tab, but the search field has a couple of options that can be accessed by clicking the magnifying glass icon.



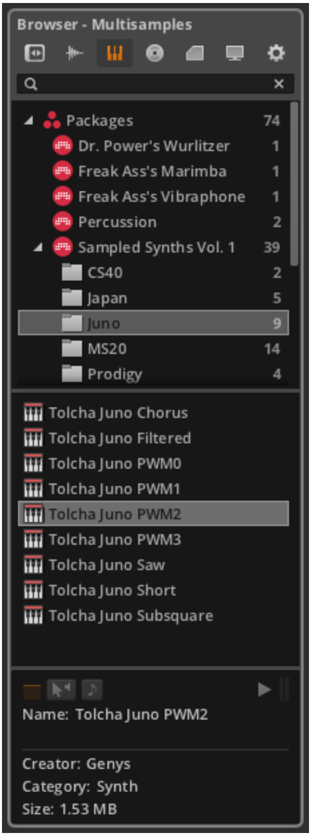
When this *ORGANIZE DEVICES BY* parameter is set to *Location*, the plug-ins will be organized by the folder they are in. When *Vendor* is selected, the plug-ins will be organized by the name of their manufacturer.

To edit the metadata for a preset: right-click the preset and then choose *Edit File Metadata...* from the context menu. Enable/disable any *Tags* in the following window, change the *Category* if you like, and select *Ok*.

4.1.2. Samples Tab

The *Samples* tab handles audio files from both the Bitwig Studio library's *samples* folder and folders that you have designated. Files found here can be loaded anywhere that audio is accepted.

4.1.3. Multisamples Tab



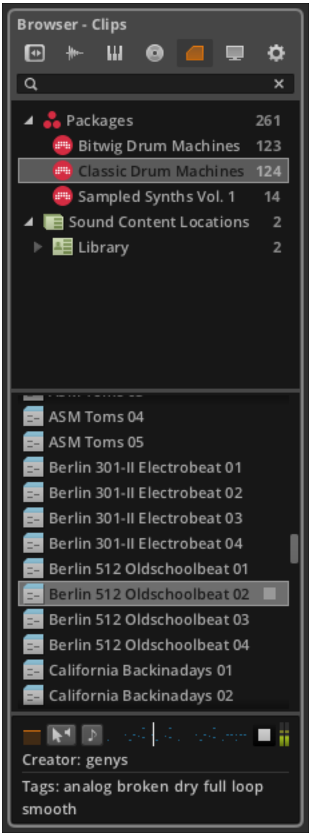
The *Multisamples* tab handles files from both the Bitwig Studio library's *multi-samples* folder and folders that you have designated. Files found here can be loaded into **Sampler** devices.

The preview function is disabled in this tab.

4.1.4. Music Tab

The *Music* tab handles recognized media files from music folders that you designate. This includes iTunes libraries. Files found here can be loaded anywhere that audio is accepted.

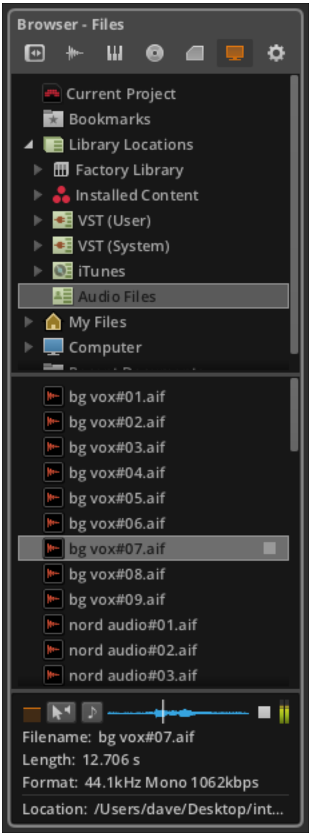
4.1.5. Clips Tab



The *Clips* tab handles files from both the Bitwig Studio library's *clips* folder and folders that you have designated. Files found here can be loaded into both the Arranger Timeline and the Clip Launcher.

To edit the metadata for a clip: right-click the preset and then choose *Edit File Metadata...* from the context menu. Enable/disable any *Tags* in the following window and select *Ok*.

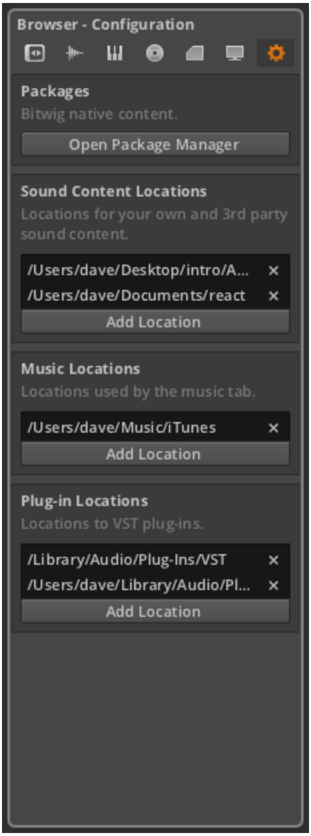
4.1.6. Files Tab



The *Files* tab allows you to navigate across the current project, any bookmarked folders or library locations, your local user folder, your entire computer, and recent files.

The search field is not present in this tab.

4.1.7. Configuration Tab



The *Configuration* tab is essentially your settings that pertain to the **Browser Panel**.

First, it gives you access to the **Package Manager**. As was discussed earlier (see section 0.1.2.1), this window allows you to install (or uninstall) additional content from Bitwig.

Second, this tab allows you to define folders on your computer that meet one of three descriptions: *Sound Content Locations*, *Music Locations*, and *Plug-in Locations*. Once you add a folder to one of these location lists, the contents of that folder will be available in the relevant Browser tab(s).

To add a folder to one of the Browser's location settings: click the appropriate *Add Location* button, navigate to the desired folder from the dialog that appears, and then click *Open*.

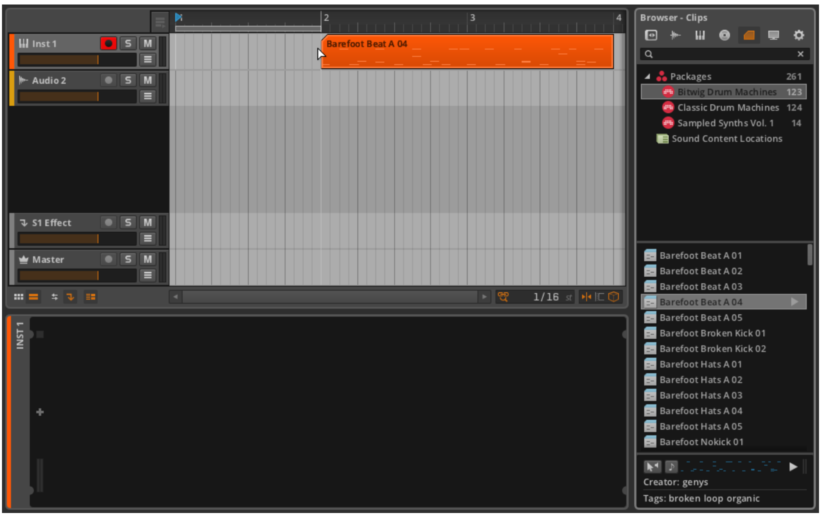
4.2. Inserting and Working with Arranger Clips

Now that we have met the **Browser Panel**, we will put it to use as a source for clips.

4.2.1. Inserting Clips

While material from several of the **Browser Panel** tabs can be inserted as clips, we will demonstrate with something from the Clips tab.

To insert a clip on an Arranger track: click and drag the clip from the **Browser Panel** to the desired timeline position on the appropriate track.



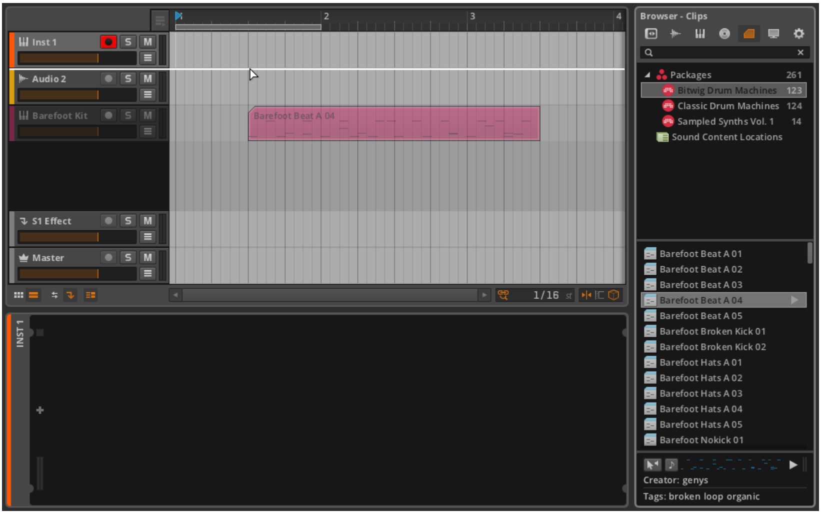
Note

Since we are dragging a note clip, it made the most sense to place it on a note track, but we could have dragged it to any track. As the concept of hybrid tracks may have indicated, Bitwig Studio is rather free with the idea of track types.

If you drag a note clip to an empty audio track, the track will be converted to an instrument track. If you drag a note to an occupied audio track, the track will be converted to a hybrid track. In both cases, the converse is true as well.

So inserting clips from the browser is as simple as dragging them into the Arranger Timeline.

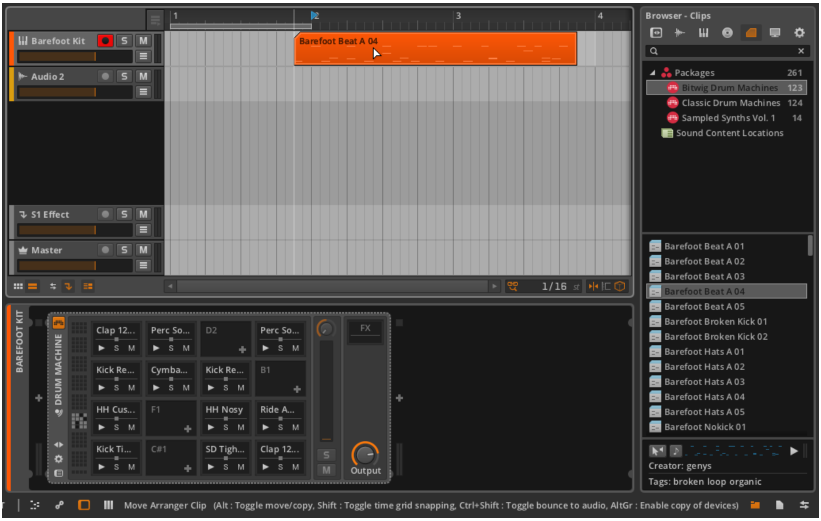
To insert a clip on a brand new Arranger track: click and drag the clip from the **Browser Panel** to the desired timeline position between existing tracks.



This method of inserting clips will work from any Browser tab whose contents can be placed on tracks. And the same method will work when dragging appropriate files from your file manager application (i.e., File Explorer on Windows, Finder on Mac, etc.) directly onto the tracks.

4.2.2. Moving Clips and Snap Settings

To move a clip within the Arranger Timeline Panel: click and drag the clip with the mouse.

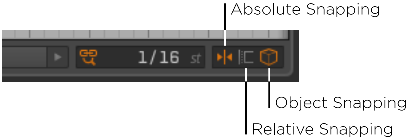


The result will be similar to when the clip was originally inserted from the **Browser Panel**. But also note that as you begin dragging the clip to move it, a status message appears in the window footer with several additional options. (This is shown in the image above; note that the order of options varies by platform, and your screen may not match the sequence in this image.)

> **Note**
>
> Do look for status messages whenever you are clicking and dragging items in Bitwig Studio. This document will not necessarily cover all variations that are shown within the program.

The first option — that adding [CTRL] ([ALT] on Mac) while dragging a selection toggles between moving and copying — was mentioned in a previous chapter.

The second option is new and indicates that [SHIFT] toggles from obeying the beat grid to ignoring it. Whether and how clips conform to the beat grid is governed by the *snap settings*, which are found in the bottom right corner of the **Arranger Timeline Panel**, just beyond the beat grid settings.



These three independent options determine which elements clips will or will not snap to as you drag them across time. As each option only provides additional anchor points, the options have no effect on each other.

› The *absolute snapping* option causes clips to snap to the current beat grid.

  This option can be toggled by either clicking on the icon or pressing [SHIFT]+[COMMA] while the **Arranger Timeline Panel** has focus.

› The *relative snapping* option uses the current beat grid resolution, but it thinks of a grid in relation to the clip's current start time. So if the clip does not start exactly on the beat grid, the amount that the clip is offset will be preserved when it is moved.

  This option can be toggled by either clicking on the icon or pressing [SHIFT]+[PERIOD] while the **Arranger Timeline Panel** has focus.

› The *object snapping* option causes clips to snap to the start and end of other clips within the Arrangement Timeline.

  This option can be toggled by either clicking on the icon or pressing [SHIFT]+[SLASH] while the **Arranger Timeline Panel** has focus.

Note

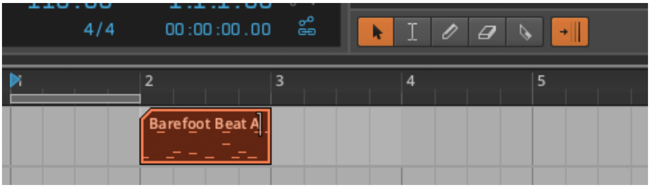
On a German keyboard, the key command is [SHIFT]+[HYPHEN].

If only one of these options is enabled, only that snapping rule applies. If multiple options are enabled, clips will momentarily snap into place for each and every rule that applies. And when none of these rules is enabled, clips will move freely, the same as when you hold down [SHIFT].

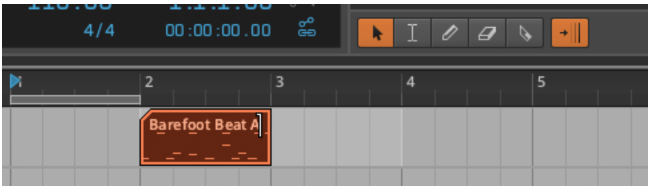
These settings will apply not just to moving clips, but to any other editing action in the panel. We will touch upon some of those actions now.

4.2.3. Adjusting Clip Lengths

To demonstrate working with the Bitwig Studio's various tools in the **Arranger Timeline Panel**, we will start with the task of removing the second half of a clip.



To shorten an Arranger clip: mouse over the top right edge of the clip so that a half-bracket cursor appears. Then click and drag to the left.

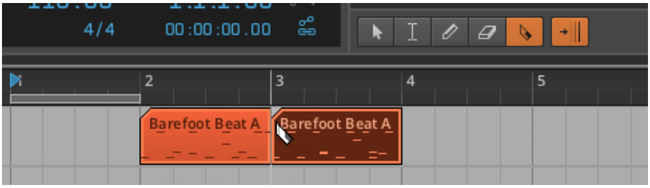


Other ways to shorten an Arranger clip include:

› With the Time Selection tool, click and drag over the time area that should be removed. Then clear the time by pressing [DELETE] or [BACKSPACE].

› With the Eraser tool, click and drag over the portion of the clip to be removed.

› With the Knife tool, click the position where the clip should be separated. Once the clip is divided, select and delete — [DELETE] or [BACKSPACE] — the unwanted clip.



All of these methods achieve the same effect. And while it may seem like the second half of our clip is now gone forever, this is not the case. Bitwig Studio still remembers the full contents of our clip in case we need it back later.

To lengthen an Arranger clip: mouse over the top right edge of the clip so that a half-bracket cursor appears. Then click and drag to the right.

Bitwig Studio acts rather nondestructively, internally preserving data whenever practicable. But you can always ask the program to stop considering data that is not currently visible by using the *consolidate* function, which essentially solidifies a clip for various purposes.

To remove unseen data from a clip: right-click the clip and then choose *Consolidate* from the context menu.

After consolidating the previous clip, extending it would now work differently.

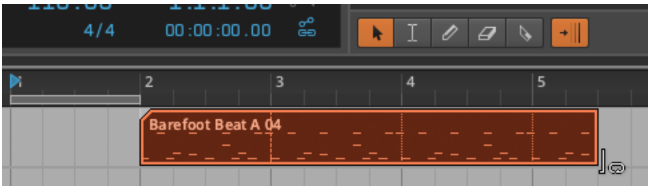
To consolidate multiple clips: select all of the clips. Then right-click one of the clips and choose *Consolidate* from the context menu.

For all of the above purposes, the consolidate function is also available by selecting *Edit › Consolidate* or by pressing [CTRL]+[J] ([CMD]+[J] on Mac).

4.2.4. Looping Clips

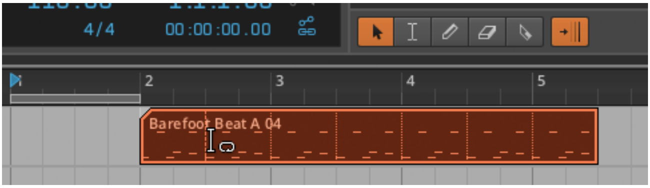
As clips are intended to be the smallest practical musical idea, you may want to loop clips.

To loop an Arranger clip: mouse over the bottom right edge of the clip so that a half-bracket cursor appears with a looping oval. Then click and drag to the right.



After you drag the clip beyond its full length, additional copies will be generated. The first copy starts with a dashed vertical line, marking the loop length being used. All subsequent repetitions of the loop are marked with dotted vertical lines. Once the clip is looping, you can do the same using any of the "bracket" tools, either at the end or beginning of the clip.

To adjust the loop length of an Arranger clip: mouse over the clip's first repeat marker (the dashed vertical line) so that an I-beam cursor appears with a looping oval. Then click and drag in either direction.

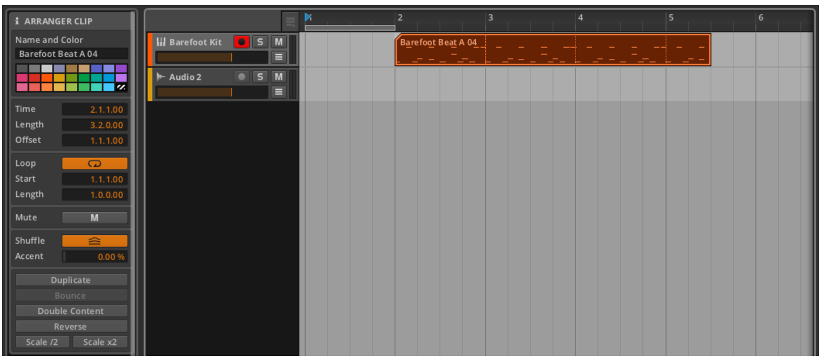


The length of the clip itself remains the same while the section of the clip that loops — and accordingly the number of repetitions — has changed.

4.2.5. The Inspector Panel on Arranger Clips

While the Arranger Timeline is a convenient, graphical view for working with the length and loop settings of a clip, all of those mouse movements are really just triggering parameter changes in the **Inspector Panel**. By investigating these parameters (along with the function buttons provided by the Inspector), we will get a clearer understanding of what is possible in Bitwig Studio in general and the Arranger in particular.

We will start by focusing the **Inspector Panel** on the same clip looping example we just finished.



For the time being, we are just paying attention to the parameters in the *ARRANGER CLIP* portion of the **Inspector Panel**. We have already seen the *Name and Color* options for tracks (see section 3.2.4). For the remaining sections, the first four offer parameters and the last one presents function buttons. The above image can be referred to throughout this discussion.

4.2.5.1. Time (Position) Section

These settings relate to the musical time or position of the selected clip:

› *Time* sets the start of the clip in the Arranger Timeline. Adjusting this position will simply move the clip exactly as it exists, the same as clicking and dragging the entire clip in the Arranger.

› *Length* sets the duration of the clip in the Arranger Timeline. Adjusting this duration will simply lengthen or shorten the clip, the same as using the bracket cursor to adjust the right edge of the clip.

› *Offset* preserves the position and length of the clip, but shifts its internal content by the set amount. This is the same as using the bracket cursor to move the left edge of the clip forward in time.

Taking the previous image as an example, I could increase the *Time* from *2.1.1.00* to *2.2.1.00*. The entire clip is now happening a quarter note later.

But if I wanted the clip to stay in time and simply skip the first beat it was playing, I would increase the *Offset* from *1.1.1.00* (no offset) to *1.2.1.00*.

Note that the first beat is included in subsequent loops.

4.2.5.2. Loop Section

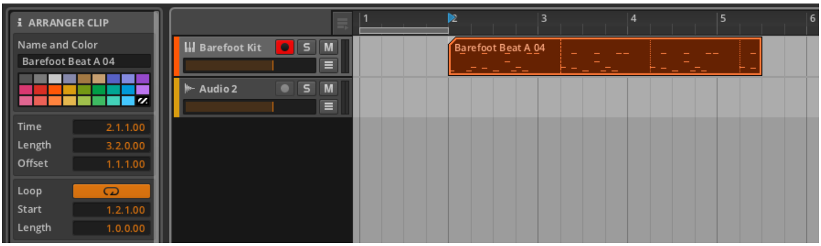
These settings relate to the looping of the selected clip:

› *Loop* toggles whether or not the clip loops with the Arranger. When disabled, the clip will play only once. If the size of the clip is longer than its contents, the later portion of the clip will be empty.

If *Loop* is off, the other settings here are ignored.

› *Start* is the looping equivalent of the *Offset* parameter, keeping the clip contents in their place but delaying the point at which each loop repetition starts.

Taking the same example from above, I could increase the *Start* from *1.1.1.00* (no loop offset) to *1.2.1.00*, causing each one-bar loop to end in the same place but start a quarter note late.



› *Length* sets the duration of the clip that is being repeated. This is the same as using the I-beam cursor with a looping oval to graphically adjust the loop length.

4.2.5.3. Mute Section

Mute toggles whether or not the selected clip is disabled on playback. This is in contrast to the track mute button, which disables all contents of the track.

4.2.5.4. Shuffle Section

These settings relate to the groove of the selected clip:

› *Shuffle* toggles whether or not the Global Groove parameters are applied to the clip. If *Shuffle* is off, the other setting here is ignored.

› *Accent* sets the percent of the Global Groove's accent *Amount* that should be applied to this clip.

For example, if the Global Groove's accent *Amount* is set to *100%* (the default setting) and the clip's *Accent* setting is at *30%*, then the clip will apply an accent at 30% strength (30% of 100%).

Or if the Global Groove's accent *Amount* is set to *50%* (the default setting) and the clip's *Accent* setting is at *50%,* then the clip will apply an accent at 25% strength (50% of 50%).

Since this is a scaling function, either parameter being set to zero (*0%*) results in no accent.

4.2.5.5. Function Buttons Section

These buttons execute the specified function on the selected clip:

› *Duplicate* places an exact copy of the selected clip immediately after it. This function is also available from *Edit › Duplicate Arranger Clip* or by pressing [CTRL]+[D] ([CMD]+[D] on Mac).

› *Bounce* prints the sound source of the selected clip into a new, solid audio clip (the functional equivalent of a "consolidated" clip). For an audio clip, the sound source is the audio itself, which will be printed into a solid clip. For a note clip, the sound source is the first instrument device in the track's device chain.

› *Double Content* makes the selected clip twice its current length and duplicates its non-looping contents.

› *Reverse* flips the order and positions of the clip's contents, causing them to play "backwards."

› *Scale /2* simply halves the length of a looping clip. This is the same result as dragging its end point exactly halfway toward the start.

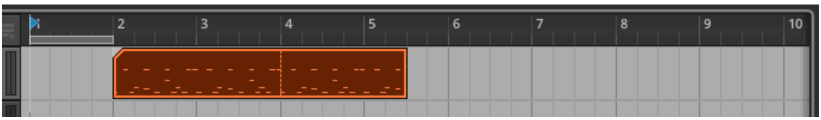
The following images demonstrate a selected looping clip both before and after the *Scale /2* function is applied:

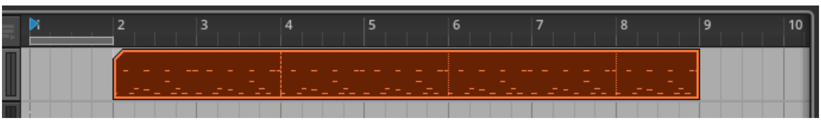
For a non-looping clip, it halves both the length of the selection and the duration and positions of its contents, effectively causing the clip to play back twice as fast.

The following images demonstrate a selected non-looping clip both before and after the *Scale /2* function is applied:

› *Scale x2* simply doubles the length of a looping clip. This is the same result as dragging its end point so that the entire clip is now twice as long.

The following images demonstrate a selected looping clip both before and after the *Scale x2* function is applied:



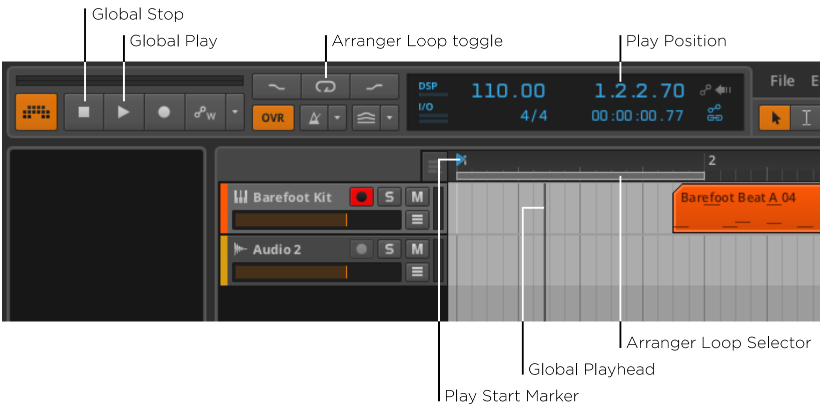


For a non-looping clip, it doubles both the length of the selection and the duration and positions of its contents, effectively causing the clip to play back half as fast.

The following images demonstrate a selected non-looping clip both before and after the *Scale x2* function is applied:

4.3. Playing Back the Arranger

How to play Arranger clips is simple enough: you play the Arranger. But there are a few details worth getting into at this point. Let's begin this discussion with the elements that enable basic playback.



To play the Arranger timeline: engage the transport by pressing either [SPACE BAR] or [P], or by clicking the Global Play button.

To stop the Arranger timeline: disengage the transport by pressing either [SPACE BAR] or [P], or by clicking the Global Stop button.

The *Global Playhead* is an indicator of where the transport has most recently played. In the Arranger Timeline, it is represented with a vertical black line. Whenever the transport is active, the Global Playhead progresses through the Arranger tracks, and its location is noted by the play position display in the window header.

The *Play Start Marker* is the blue, right-facing triangle within the Beat Ruler that indicates where the transport will play from the next time it is engaged.

To move the Play Start Marker: single-click in the top half of the Beat Ruler.

Other ways to move the Play Start Marker include:

› Single-click anywhere within the Arranger Timeline with the Object Selection tool.

› Click and drag the play position in the window header's display section.

› Select a single Arranger clip to move the Play Start Marker to the beginning of that clip.

To play the Arranger timeline from the beginning: press [ALT]+[SPACE BAR] or [ALT]+[P].

To play the Arranger timeline from the Global Playhead's position: press [SHIFT]+[SPACE BAR] or [SHIFT]+[P].

To stop the Arranger timeline and advance the Play Start Marker: click the Global Play button.

The *Arranger Loop Selector* sets the region of the Arranger Timeline that will be looped during playback. This region is also used for several other functions.

To toggle the Arranger Loop function: click the Arranger Loop toggle in the window header.

The Arranger Loop function affects all tracks as it literally picks up and moves back the Global Playhead when the end of the region is reached. This is a playback function, while clip looping is an arrangement function.

To move the Arranger Loop Selector's position: click the center of the Arranger Loop Selector and drag it in time.

To change the Arranger Loop Selector's length: mouse over the left or right edge of the Arranger Loop Selector so that a bracket cursor appears. Then click and drag in either direction.

4.3.1. Cue Markers

You also have the option of using *Cue Markers* in the Arranger, which store play positions along the Arranger Timeline for easy triggering. To use Cue Markers, first select *View › Show Cue Markers* or press [ALT]+[SHIFT]+[C]. This will make the Beat Ruler slightly taller.

To create a Cue Marker: double-click the bottom of the Beat Ruler.

The right edge of a Cue Marker's play button icon aligns with its location.

To trigger playback from a Cue Marker: double-click its play icon.

If the transport was inactive, playback will start immediately from the Cue Marker. If the transport was already going, playback will move to the Cue Marker's position after the *Default Launch Quantization* interval (see section 5.2.3.2).

Note

If you want the same playback behavior without creating a Cue Marker, simply double-click the desired playback position from the top of the Beat Ruler (between the numbers).

To rename a Cue Marker: double-click its name.

To change a Cue Marker's color: right-click either the Cue Marker's icon or name, and then select a different color from the palette that appears within the context menu.

To move a Cue Marker: click either the Cue Marker's icon or name, and then drag it to the desired position. Or click the Cue Marker to select it, and then change its position in the **Inspector Panel**.

To delete a Cue Marker: click the Cue Marker to select it, and then press [DELETE] or [BACKSPACE].

4.4. Recording Clips

Since we can now edit Arranger clips in the most fundamental ways, it is a good time to examine recording new note and audio clips. This begins with getting the right signals routed into our tracks.

Before we deal with this on a track level, make sure that any audio and MIDI interfaces/controllers you are using have been set up properly (see section 0.1.2).

4.4.1. Track I/O Settings

To assign input and output paths for each track, we must first have access to the Track I/O section within each track header. This section's visibility is toggled by either clicking the Track I/O view toggle or selecting *View › Show Track I/O*.

This section contains the following controls:

› The *input chooser* lets you select which signals are getting routed into the track.

For instruments tracks, the options are incoming MIDI sources. The default selection is *All inputs* so that every MIDI source should make it to the track.

For audio tracks, the options are both incoming audio sources and the audio outputs of all other tracks. The default selection is *No input*.

› The *output chooser* lets you select where the track's final audio is getting routed to. The default selection is *Master*, which will serve us well in nearly all situations.

› The *monitor button* toggles whether the selected input source is being passed to the track's input. All instrument tracks have monitoring enabled by default.

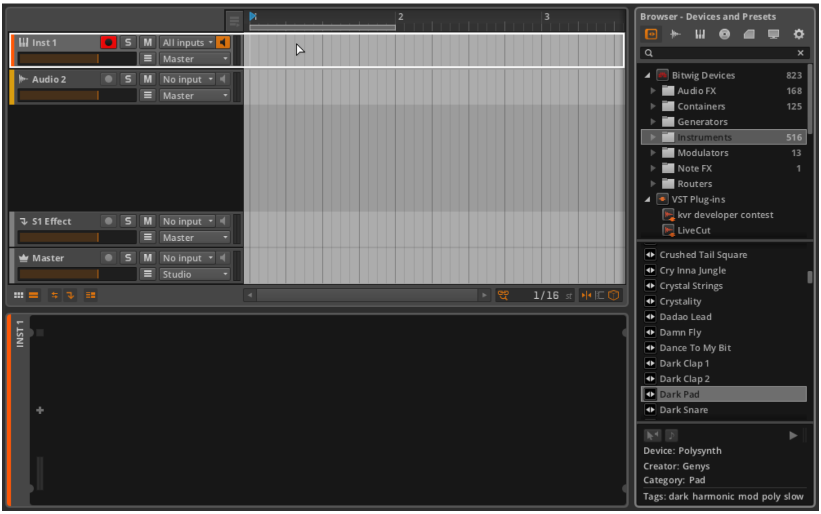
4.4.2. Recording Note Clips

A few steps are needed to successfully record a note clip. First, we need a sound source for our audio. Second, we need a MIDI source to record as notes. And then, we can hit record.

4.4.2.1. Loading an Instrument Preset

Note clips in Bitwig Studio — not unlike MIDI — are really just instructions to be interpreted by an instrument device. Notes themselves do not produce any sound. So before we record any notes, we should load an instrument device so that our notes can be realized.

To load an instrument device: go to the **Browser Panel** and select the Devices and Presets tab. Under *Bitwig Devices*, select the *Instruments* folder. From the selection pane, drag any preset into the **Arrangement Timeline Panel**.



If you do not like the first device preset you load, repeat the above steps until you find one you appreciate.

4.4.2.2. Setting a MIDI Source

If you have a MIDI keyboard connected and already made Bitwig Studio aware of it, then it should be working already. By playing the keys, the instrument track's level meters should start showing audio.

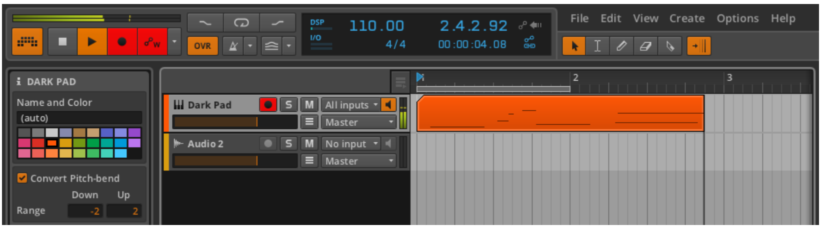
If you do not have a MIDI controller — or your MIDI device is all knobs and no keys — press [CAPS LOCK] to temporarily transform your computer keyboard into a MIDI keyboard. Pressing letters in the top two rows should trigger notes and cause the audio meters to dance.

Note

While [CAPS LOCK] is active, most key commands will not work.

4.4.2.3. Recording Notes

To record an Arranger note clip: enable the track's record arm button, enable the Global Record button, and then activate the transport and begin playing notes.



4.4.3. Recording Audio Clips

Unlike notes, the audio events that make up audio clips do not require any devices. They are already audio. So once we determine the audio source to be recorded, we should be good to go.

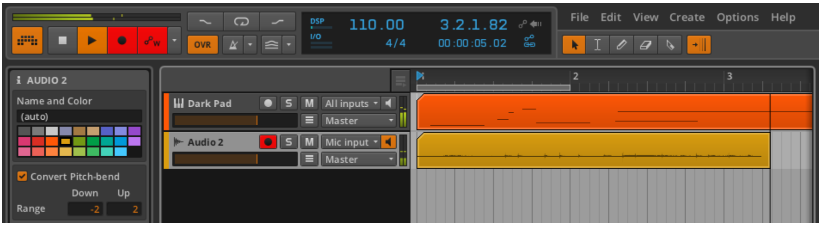
4.4.3.1. Setting an Audio Source

Whether you are using an external audio interface or the internal interface of your computer, you first need to set the desired input source in the track's input chooser (they will be at the top of the chooser list). If you enable the monitor button of the track and then send audio to this input, you should see the input in the track's audio meters.

Before recording, you probably want to disable the record arm buttons on all other tracks. Otherwise, you could trigger multiple tracks to record at once and alter or erase other clips in the process.

4.4.3.2. Recording Audio

To record an Arranger audio clip: enable the track's record arm button, enable the Global Record button, and then activate the transport.



5. The Clip Launcher

We have spent the last couple chapters working within the Arranger Timeline. And while the Arranger is absolutely crucial to music creation in Bitwig Studio, it is only half of the story.

The **Clip Launcher Panel** — also called the *Launcher* — is the logical Arranger's artistic brother. While the Arranger is an excellent way to lay out the fixed "story" of a song, the Launcher allows you to freely improvise with your clips. More on that soon.

We will start by getting an overview of the **Clip Launcher Panel** and its constituent elements. Next we will revisit some of the same concepts we saw with Arranger clips as they apply to Launcher clips. We will then investigate how Launcher clips relate to the transport and Arranger clips and see how Launcher clips are triggered. Finally, we will record Launcher clips and learn to capture the Clip Launcher's output on the Arranger Timeline.

Bitwig Studio is just one DAW, but it is the two sequencers within that provide limitless musical possibilities.

5.1. The Clip Launcher Panel

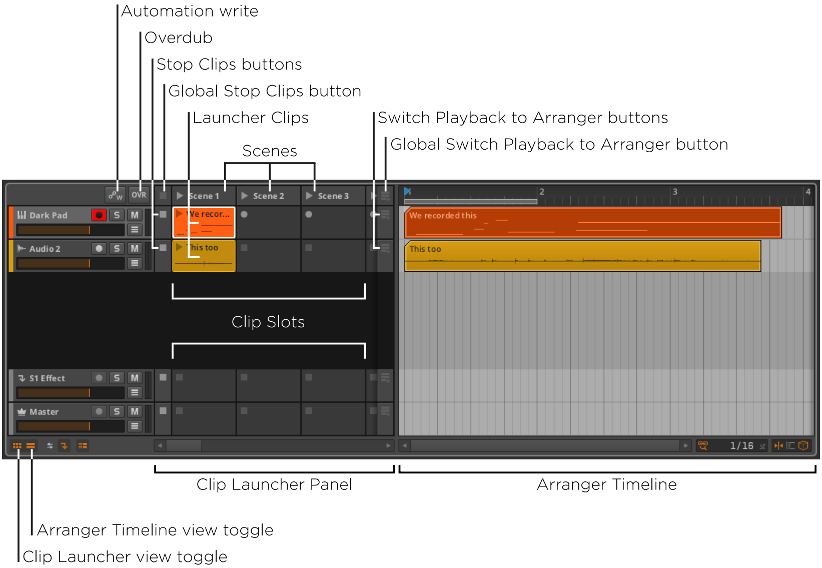
Charting out music from beginning to end is the way nearly all productions take place. But even from the earliest music, improvisation has been an important source of variation, inspiration, and life. Balancing these two poles — the programmed and the spontaneous — has been a central concern, all the way from Bach's time and his (literally) sacred music, up to the present day and our attempts to make electronic music engaging from the stage.

Aside from its unique perspective and purpose, the **Clip Launcher Panel** is also the only panel that loads directly into another panel. In this chapter, we will be learning about the Launcher within the **Arranger Timeline Panel**, but it can also be called up inside the **Mixer Panel** of the **Mix View** (see section 6.1.2).

The key difference between Arranger clips and Launcher clips is their purpose. Arranger clips are played back precisely at the designated time. But Launcher clips must be available whenever you want them, either for section-based composition (verse, chorus, bridge), or as pieces for a live performance, or however else you might use them. Arranger clips must be rigid, and Launcher clips must follow your whim.

5.1.1. Clip Launcher Layout

Let's begin by examining the **Clip Launcher Panel** beside the Arranger Timeline that we were just using.



What we see here is the same **Arranger Timeline Panel** as before, but now the view toggles for both the Clip Launcher and the Arranger Timeline are engaged. As a result, we see these two sequencers side by side within the panel.

The **Clip Launcher Panel** appears as a series of *slots* that are arranged across each track. Since tracks in the **Arrange View** are oriented horizontally, the **Clip Launcher Panel** is also arranged from left to right. In case more slots exist than can be shown at one time, the horizontal scroll bar at the bottom of the panel allows you to scroll to view all the slots.

The slots are made to house clips and have no functionality of their own. Whenever we refer to a "Launcher clip," we mean a clip that is housed within this Launcher sequencer.

On each track before the clip slots begin is a *Stop Clips button*. Each of these buttons halts all clips that were playing on its track. And on each track after the last visible clip slot is a *Switch Playback to Arranger button*. Each of these buttons restores the Arranger as the active sequencer for this track. The last section of this chapter will explain this relationship in detail.

Each vertical column of clips is a group called a *scene*. These groupings can be used for triggering or working with the constituent clips all together. If additional slots are needed, additional scenes can be created to provide them.

Similar to each track, the displayed scenes begin and end with the *Global Stop Clips button* and the *Global Switch Playback to Arranger button*, respectively. Each global button is the equivalent of triggering all track buttons of that kind. Again, the last section of this chapter will cover these functions in more detail.

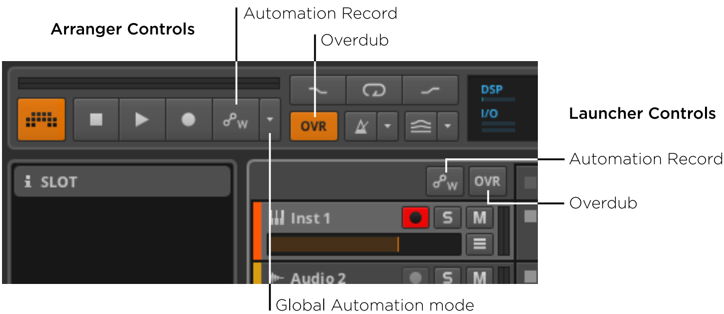
Finally, to the left of the scenes (just above the track headers) are a couple of global controls for the **Clip Launcher Panel**.

› *Automation Record*: Enables automation recording to the **Clip Launcher Panel**.

› *Overdub*: Merges incoming notes onto active clips of all record-enabled tracks the next time the transport is started. Otherwise, note data is overwritten.

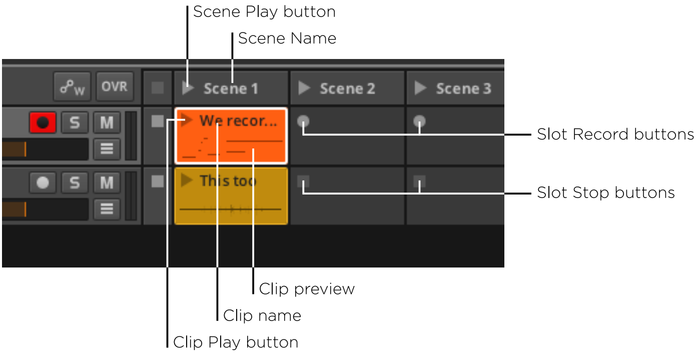
Note

Just as these controls only affect the **Clip Launcher Panel**, the similar options in the window header's transport controls section affect only the Arranger Timeline. The Automation Record mode setting, however, applies to both the Arranger and Launcher (see section 8.1.5).



5.1.2. Within Launcher Clips, Scenes, and Slots

As for the appearance of Launcher clips themselves, there are only a few things to note.



The crucial item within each clip and scene is the *play button*. This is the means by which you trigger the clip or scene. These play buttons also serve as indicators of which clips are active.

The top of each clip and scene also leaves space for that item's *name*, which is optional. As can be seen in the image above, scenes without names may be given automatic ones which you can always replace manually.

Below the play button and name of a clip may be a *preview* of the clip's contents. Clips that contain either notes or audio events will always have a preview, but the preview can be shown only when the track height is set to normal. When the **Arranger Timeline Panel** has tracks set to half size (as shown below), there is no room for the preview.

Finally, a couple of different buttons can appear within empty slots.

If the track is record-enabled, a *slot record button* will appear about where the play button would within a clip. Clicking this record button activates recording within the clip.

If the track is not record-enabled, a *slot stop button* will appear instead. This button is just an alias to the track's Stop All clips button, performing the exact same function.

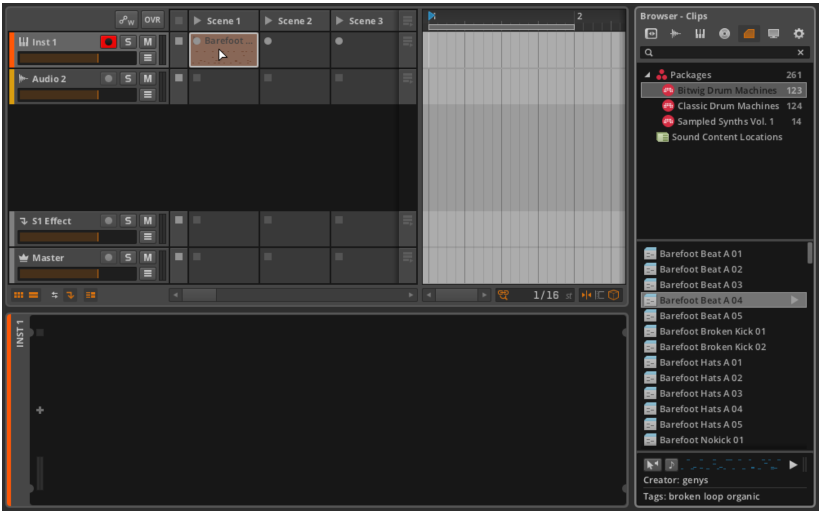
5.2. Acquiring and Working with Launcher Clips

Before manipulating clips in familiar ways, we must first get clips into the Launcher. We will start by recapping inserting and recording clips, and then look at moving clips between the Arranger and Launcher. Finally, we will see how length and looping adjustments are handled in the **Clip Launcher Panel** with the help of the **Inspector Panel**.

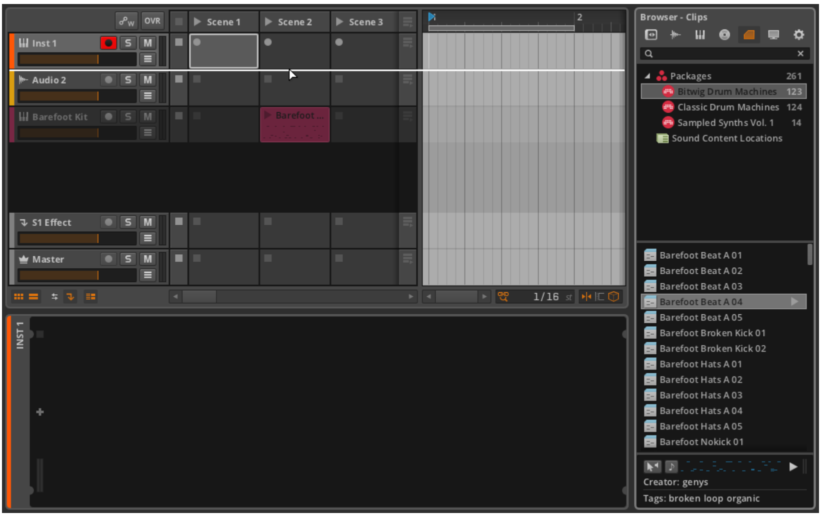
In the **Clip Launcher Panel**, we will recap inserting clips from the **Browser Panel**, look at moving clips between the Launcher and the Arranger, and see the options available for Launcher clips in the **Inspector Panel**.

5.2.1. Getting Clips from the Browser Panel

Getting clips from the **Browser Panel** onto a track is almost identical for the **Clip Launcher Panel** and the Arranger Timeline (see section 4.2.1). The only difference is where you drop the clip off.



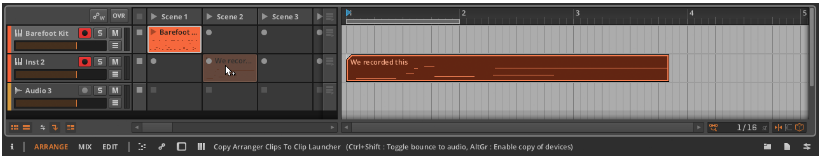
And if the clip is dragged between two tracks, a new track will be created automatically as well.



5.2.2. Copying Clips between the Arranger and Launcher

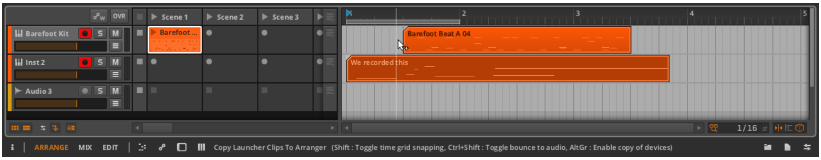
Copying a clip from one sequencer to the other follows the same pattern as all other movements that we have made.

To copy an Arranger clip to the Launcher: click and drag the clip from the Arranger Timeline to the desired slot on the appropriate track.



If multiple Arranger clips are selected, the clips will be copied into successive slots.

To copy a Launcher clip to the Arranger: click and drag the clip from the Launcher to the desired timeline position on the appropriate track.



If multiple Launcher clips are selected, the clips will be placed into the Arranger consecutively.

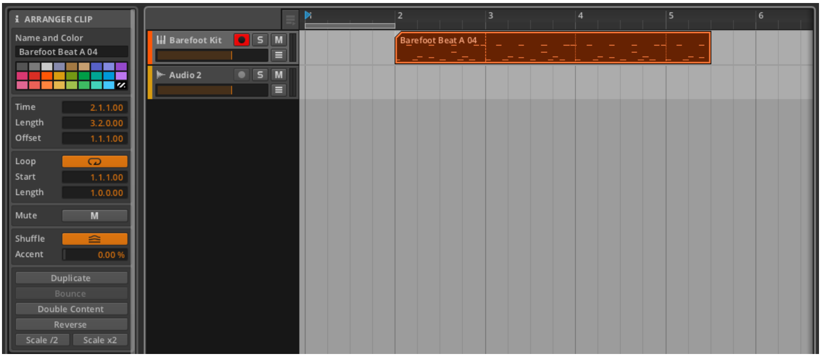
Scenes can also be copied from the Launcher to the Arranger Timeline. And conversely, any combination of Arranger clips can be copied to a scene by dragging them over.

All of these copy functions can also be done into new tracks.

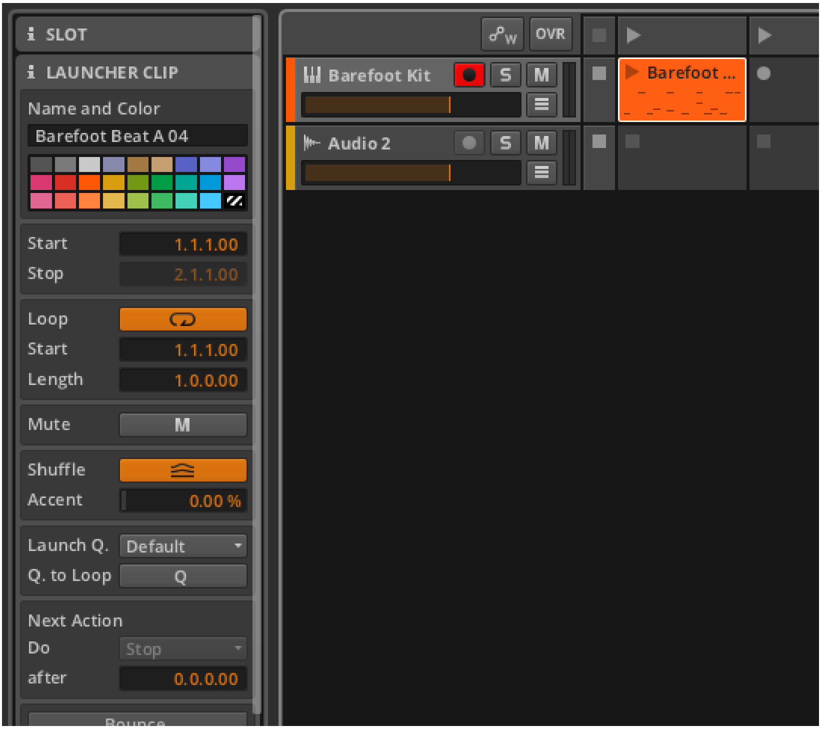
5.2.3. Launcher Clip Parameters

The Arranger Timeline had a convenient graphical view for visualizing the length and loop settings of a clip. While the **Clip Launcher Panel** does not have its own graphical editor, we will always have the **Inspector Panel**.

Launcher clip parameters are generally similar to Arranger clip parameters with a few important differences. In order to see how the **Inspector Panel** represents Launcher clip information, let's revisit the Arranger clip looping example from the last chapter.



In this case, I have copied the example Arranger clip into a Launcher slot. The resultant Launcher clip gives us these settings in the **Inspector Panel**.



We can see that the *Loop*, *Mute*, and *Shuffle* sections are identical to their Arranger clip counterparts, and that we have also seen all the functions available here already (see section 4.2.5).

We can also see that the initial *Start/Stop* section differs from the Arranger clip's *Time* (Position) model, and that the *Launch Q.* and *Next Action* sections are completely new.

5.2.3.1. Start/Stop Section

Arranger clips had the *Time* (Position) section because they are always triggered at the exact position where they reside. Since Launcher clips do not share this sense of predestination, their parameters simply describe what portion of the clip should be played when triggered.

This section contains the following controls:

› *Start* sets the location within the clip that should be played first. This is very similar to adjusting the *Offset* of an Arranger clip, changing only which part of the Launcher clip should play back first.

› *Stop* sets the end of the clip contents that should be played. This setting is available only when *Loop* is disabled.

5.2.3.2. Launch Q(uantize) Section

Launch quantization is a crucial concept for getting the Launcher to play in time with the Arranger.

Clips are usually triggered with a performance gesture, such as a click of the mouse or the push of a button. Since musicians are not robots, these gestures are almost always imprecise, landing somewhere other than the exact beat we meant to hit.

Launch quantization forces the clips we trigger to align appropriately with the beat grid. Since things that have already started cannot be shifted backwards in time, we must trigger the clips ahead of time so they can match the next defined beat exactly. (You can think of launch quantization as a performance-based version of absolute grid snapping.)

This section contains the following controls:

› *Launch Q.* sets the interval at which this particular clip will be triggered.

A beat-level setting (for example, *1/2*, *1/4*, *1/8*, or *1/16*) will play all newly triggered clips when the Global Playhead reaches the next grid line of that interval.

A bar-level setting (for example, *1 bar*, *2 bars*, *4 bars*, or *8 bars*) will play all newly triggered clips when the Global Playhead reaches the next bar of this interval. For example, a setting of *1 bar* would wait for beat 1 of the next bar to play, while a setting of *4 bars* would wait for the next fourth bar (e.g., bar 1, bar 5, bar 9, etc.) to be reached.

Off disables clip quantization, meaning the clip will begin playback the moment it is triggered.

Default follows the *Default Launch Quantization* setting, which is defined under *Options* › *Default Launch Quantization*. This global parameter is selected from the same values listed here.

› *Q. to Loop* toggles clip quantization to be based on the loop start point instead of the clip start. This allows you to trigger a clip with a lead-in that plays once, like musical pick-up notes.

5.2.3.3. Next Action Section

Next Action is the option to determine what should happen after this clip has played for a set amount of time. The two parameters used to achieve this are *Do* and *after*, as in, "please *Do* <this function> *after* <this amount of musical time has passed>."

If *after* is set to no time (*0.0.0.00*), the Next Action function is disabled for this clip. When *after* is set to some amount of time, the clip will be played for this amount of time before the *Do* function is triggered.

Do sets the task to be executed. The following functions are available:

› *Stop* simply stops the clip.

› *Return to Arrangement* resumes playback of the Arrangement Timeline for this track.

› *Return to Last Clip* resumes playback of the Launcher clip that was playing immediately before this one. If no clip was playing when this one was triggered, the clip is stopped.

› *Play Next* triggers the next available Launcher clip. If this is the last clip on the track, the clip is stopped.

› *Play Previous* triggers the previous available Launcher clip. If this is the first clip on the track, the clip is stopped.

› *Play First* triggers the first Launcher clip on the track.

› *Play Last* triggers the last Launcher clip on the track.

› *Play Random* triggers a Launcher clip from the track at random, which could potentially retrigger this clip.

› *Play Other* triggers a different Launcher clip from the track at random. This clip will not be retriggered.

› *Round-Robin* triggers the next available Launcher clip. If this is the last clip on the track, the first Launcher clip on the track is triggered.

5.3. Triggering Launcher Clips

Just as the last chapter looked at playing the Arranger and its clips, we should now discuss triggering Launcher clips. But now that we have two sequencers in play, we must first discuss the relationship between the Arranger and Launcher. Understanding their alliance will allow you to get the most — and possibly most interesting results — out of Bitwig Studio.

5.3.1. How the Arranger and Launcher Work Together

When thinking about Bitwig Studio's two distinct sequencers, it helps to consider the following concepts:

- The transport drives all timing functions, whether it is the playback of Launcher clips, the recording of Arranger clips, or vice versa.

- The Arranger Timeline's Beat Ruler also has influence over the **Clip Launcher Panel**. Launcher clips may be played back whenever you choose, but the launch quantize feature described above is regularly used for the sake of coherence and musicality, aligning launched clips with arranged ones according to your wishes.

- On each individual track, either the Launcher or Arranger will be active at any given time.

- By default, each track starts with the Arranger Timeline active. The Launcher will take over for a track after a Launcher clip is either triggered or recorded, or the track's Stop Clips button is pressed. The Arranger will regain control only after the track's Switch Playback to Arranger button is pressed.

- All tracks can be toggled in unison from the Arranger to the Launcher and back. The Launcher will take over all tracks when either the Global Stop Clips button is pressed or a scene is triggered. The Arranger will regain control of all tracks when the Global Switch Playback to Arranger button is pressed.

The takeaway is that you can act like Bitwig Studio has just one sequencer, by using only the Arranger Timeline (to create a completely composed song, for example) or only the Clip Launcher (to take elements you have made and freely improvise a structure). You could also keep most tracks playing what you programmed in the Arranger, and occasionally shift some tracks to the Launcher for the sake of improvisation.

Once the two sequencers make sense to you, there is no "right way" to use them. Only options.

5.3.2. Triggering Launcher Clips

To trigger a Launcher clip: click the play button in its top left corner.

If the transport was stopped, triggering a clip immediately activates the transport. (Otherwise, no clip could play.)

Once a clip is triggered, a black box appears around the play button to mark this as an *active clip*. A clip remains active until either a different clip on that track is triggered, the track's (or the Global) Stop All Clips button is triggered, or the track's (or the Global) Switch Playback to Arranger button is pressed. When the transport is activated, all active clips resume playing.

In the image above, you may also notice a vertical line going through the active clip. Each active clip has its own *clip playhead* that indicates the play position within the clip while the transport is active.

To trigger a scene: clip the play button in its top left corner.

This will trigger all clips that exist within the scene and Stop All Clips for tracks that contain no clip for the scene.

To stop all clips on a track: click either the track's Stop All Clips button or a stop button within an empty slot.

This stops Arranger clips as well since the Launcher is given control of the track. Each Stop All Clips button will take effect at the default launch quantize interval.

To stop all clips: click the Global Stop All Clips button.

While this will stop all clips after the default launch quantize interval, the transport remains active.

To return control of a track to the Arranger: click the track's Switch Playback to Arranger button.

This will take effect immediately, regardless of the default launch quantize setting.

To return control of all tracks to the Arranger: click the Global Switch Playback to Arranger button.

This will take effect immediately, regardless of the default launch quantize setting.

5.4. Recording Launcher Clips

We will finally return to recording with the Clip Launcher, both to record new Launcher clips and to print the results that come out of the Launcher.

5.4.1. Recording Clips

All the same requirements apply for recording Launcher clips as Arranger clips (see section 4.4).

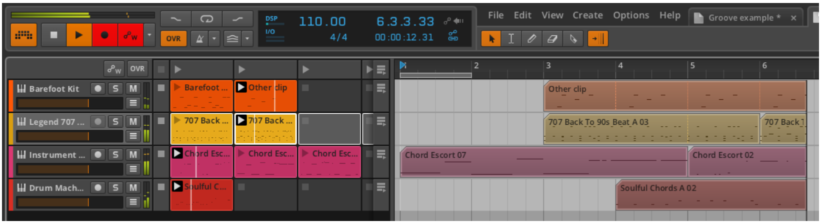
To record a Launcher note clip: enable the track's record arm button, click a blank slot's record button, and then begin playing notes.

If the transport was inactive, it will automatically start once you click the slot record button. If the transport was already active, it will continue moving, and recording will commence after the default launch quantize interval.

5.4.2. Record to Arranger Timeline

As one more form of interaction between the Launcher and Arranger, the result of all triggered Launcher clips can be recorded directly to each Arranger track. This is a way to capture an improvisation, whether from an early production phase, a stage performance, or whatever else you can imagine.

To capture clips and/or scenes triggered from the Launcher into the Arranger: enable the Global Record button, activate the transport, and then trigger the clips/scenes.



A few notes that may be helpful here.

› If you activate the transport by triggering your first clip or scene, recording will begin cleaning from the Play Start Marker.

› If you deactivate the record arm buttons of individual tracks, you will avoid recording empty clips to the Arranger tracks.

› Control changes can also be captured, making for a fully editable transcription.

6. The Mix View

For the past three chapters, we have dealt exclusively with the **Arrange View**, and more particularly with the **Arranger Timeline Panel** that is housed there. And while we are not done with the **Arrange View** (it will be back), it is time to see another of Bitwig Studio's views.

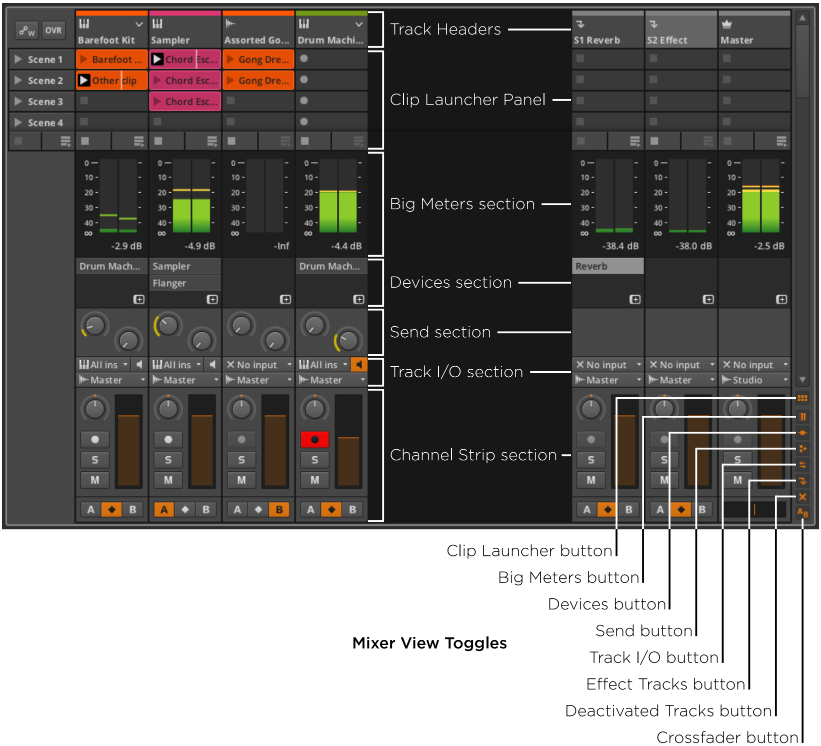
In this chapter we will take up the **Mix View** and its central **Mixer Panel**. As the purpose of each view is to provide tools organized around a musical task, the clear task of this view is *mixing*, the necessary art of adjusting and blending your tracks so that they play well together. This happens first at the master track, and then on to the real world, in headphones and on speakers.

We will begin by taking stock of the **Mixer Panel**, examining various functional details along the way. We will also look at places outside of the **Mix View** where mixing functions crop up. Finally, we will see how the master track's output can be easily controlled with the **Studio I/O Panel**.

6.1. The Mixer Panel

We will begin our examination with the **Mixer Panel** itself. Within the **Mixer View**, the **Mixer Panel** is the lone central panel.

The **Arranger Timeline Panel** was oriented horizontally, which is perfectly sensible for viewing the left-to-right timeline of your music. Just as sensibly, the **Mixer Panel** is laid out vertically like a traditional mixing board, with each available section stacked one atop the other.

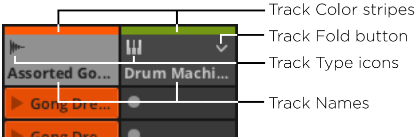


The first and next to last sections (track headers at top, channel strip sections near the bottom) will always be visible. The *View Toggles* on the bottom right allow you to decide whether each of the six other sections are shown or hidden, with another two options for whether the effect tracks or deactivated tracks should be displayed.

We will take the sections of the **Mixer Panel** in order, starting at the top.

6.1.1. Track Headers

The *track headers* in the **Mixer Panel** contain the same information as the track headers of the **Arranger Timeline Panel**.



Each track header consists of at least three parts:

› *Track Color stripe*: A swatch of the track's assigned color.

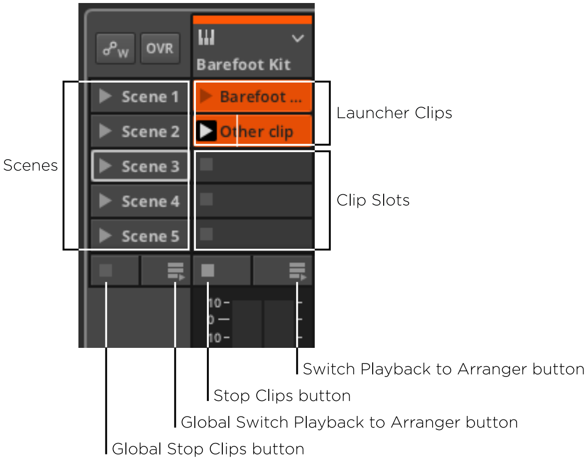
› *Track Type icon*: An icon to indicate the kind of track (see section 3.2.1).

› *Track Name*: The title assigned to the track.

› *Track Fold button:* Available for tracks whose primary signal path includes certain container devices (such as **Drum Machine**, **Instrument Layer**, or **FX Layer**). When enabled, the track's channel strip expands to the right, exposing all signal paths in the top-level of the container and giving each its own channel strip.

6.1.2. Clip Launcher Panel

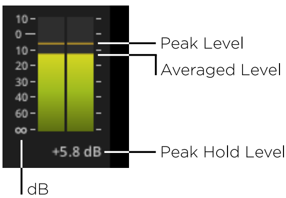
The **Clip Launcher Panel** contains all of its usual elements and functionality when loaded into the **Mixer Panel** (see chapter 5).



Its elements have just been rearranged to fit the vertical orientation of tracks in this view.

6.1.3. Big Meters Section

These high-resolution stereo audio meters — aka the *big meters* — show each track's current output level.



The meters themselves display two values:

› The top of the filled bars indicates the current *average level* (roughly, the track's current power).

› The dashed line above the bars indicates the momentary *peak level*.

Along the left side of the meters are units of decibels (dB). The values at bottom are negative (beginning with negative infinity), eventually rising to zero, and ending in positive decibels at the top.

Beneath each meter on the right is a running record of the track's *peak hold level*.

To reset a meter's peak hold level: click on the peak hold level.

The meters section will take up as much window space as is available so hiding other sections will grow these meters and enhance their resolution.

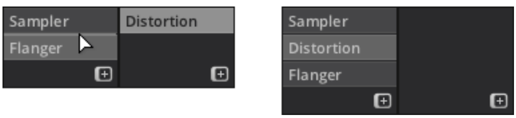
6.1.4. Devices Section

The *devices section* provides a list of all the top-level devices on each track.

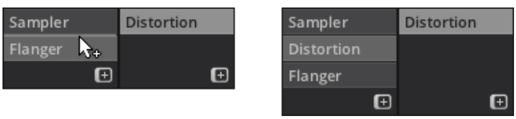
This is not to be confused with the **Device Panel** (see section 7.2), where parameters can be accessed and edited. This section can be used to call up the **Device Panel**, move/copy the devices present, and add new devices.

To focus on a track's device within the Device Panel: double-click the device.

To move a device: click and drag the device to the desired location.



You can also hold [ALT] to copy the device.

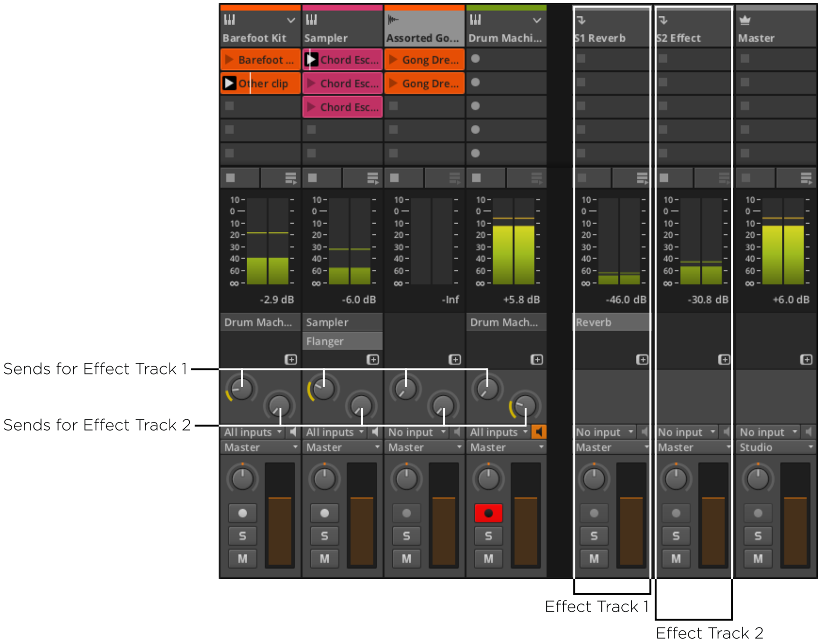


To layer a device with another: [SHIFT]-click and drag the device over top of the device where the layer should be inserted.

To add a device: either click the track's *add device button* or right-click anywhere in the device section to pull up the *Add Device window*. Then click any device from this window to add it.

6.1.5. Send Section

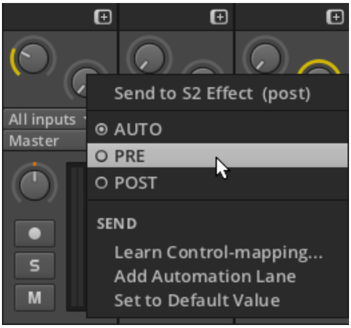
The *send section* provides a level knob for each effect track in your project. This section is available only on instrument, audio, and hybrid tracks.



Sends allow us to pass a portion of each track's audio into the various effect tracks. Using a send does not affect a track's main output level.

For each individual send, you can decide whether the audio being sent is taken before the track's volume fader has been applied or after. Since this setting is relative to the track's fader, the settings are called *pre* (for pre-fader) and *post* (post-fader). A third choice of *auto* is selected by default, permitting the effect track targeted to decide whether *pre* or *post* should be used (see section 6.2.3).

To set a send's source setting: right-click the send, and then select the appropriate setting from the context menu.



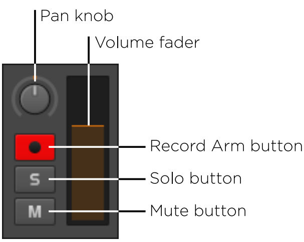
Note that the indicator ring of *post* sends is colored yellow. This same ring is colored blue for *pre* sends.

6.1.6. Track I/O Section

The *Track I/O section* allows you to assign the input and output paths for each track. This is exactly the same as it appears in the **Arranger Timeline Panel** (see section 4.4.1).

6.1.7. Channel Strip Section

The *channel strip section* contains most of the same control items as the track headers of the **Arranger Timeline Panel**.



This section contains the following controls:

› *Pan knob:* A stereo placement control for the track.

› *Record Arm button*: Record enables the track.

› *Solo button*: When any track has its solo button enabled, only tracks with solo enabled will output their audio.

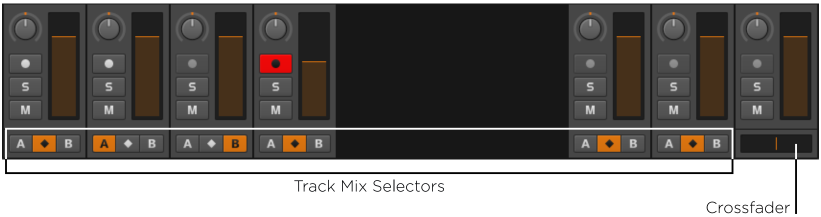
› *Mute button*: Disables the track's audio output.

› *Level meters*: Stereo audio meters that display the track's output level.

› *Volume fader*: A final level control for the track.

6.1.8. Crossfader Section

The *crossfader section* contains a *Global Crossfader* on the master track. Every other track has a *Track Mix Selector*, which allows you to designate whether that track belongs to the *A* mix, both mixes, or the *B* mix, respectively.



› When a track mix selector is set to the *A* position, that track will be unaffected when the global crossfader is anywhere between the leftmost and center positions, but that track's level will be gradually faded out as the global crossfader moves from the center position to the far right.

› When a track mix selector is set to the *B* position, that track will be unaffected when the global crossfader is anywhere between the rightmost and center positions, but that track's level will be gradually faded out as the global crossfader moves from the center position to the far left.

› When a track mix selector is set to the both mixes option (the diamond button at center), that track is completely unaffected by the global crossfader.

Note

Realize that the crossfader settings are active regardless of whether the crossfader section is visible or not.

6.2. Other Mixing Interfaces

While the functions offered by the **Mix Panel** within the **Mix View** are extensive, a subset of these options can be found both in the secondary **Mixer Panel** and within the **Inspector Panel** when tracks are selected.

6.2.1. The Secondary Mixer Panel

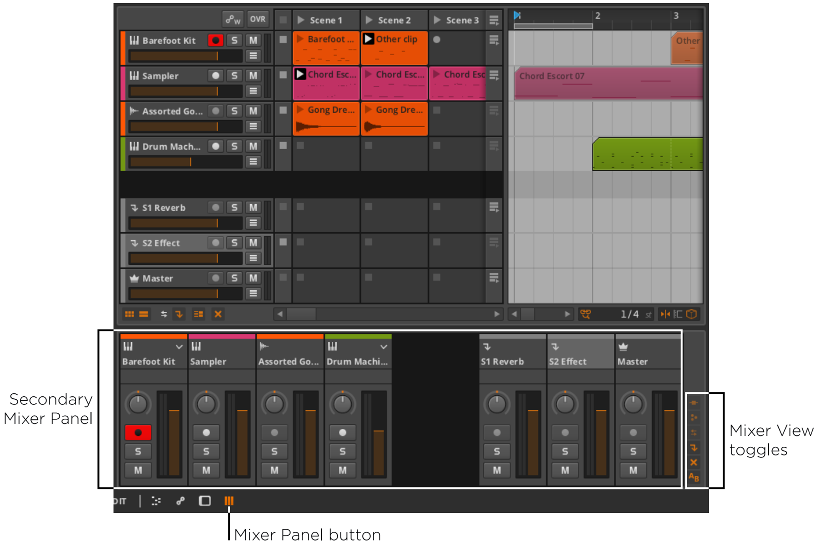
Unlike the **Arranger Timeline Panel**, the **Mixer Panel** can be loaded as a secondary panel in other views. We will briefly examine this version of the panel within the **Arrange View**.

To load the secondary Mixer Panel: click the **Mixer Panel** button in the window footer, or press either [M] or [ALT]+[M].

Note

Not every view supports every panel. The available panels within a particular view will have their buttons shown in the window footer.

For a review of these buttons and how to load the various panels, see section 2.2.1.



Again the bottom right of the panel includes the Mixer view toggles. But while all the toggles appear enabled, there are curiously few sections being displayed.

By looking closer at the view toggles, you will notice that they are all enabled but grayed out. Bitwig Studio is acknowledging that you have these sections enabled, but is also letting you know that there isn't enough vertical space to display them all. While not all panels are resizable, this one is.

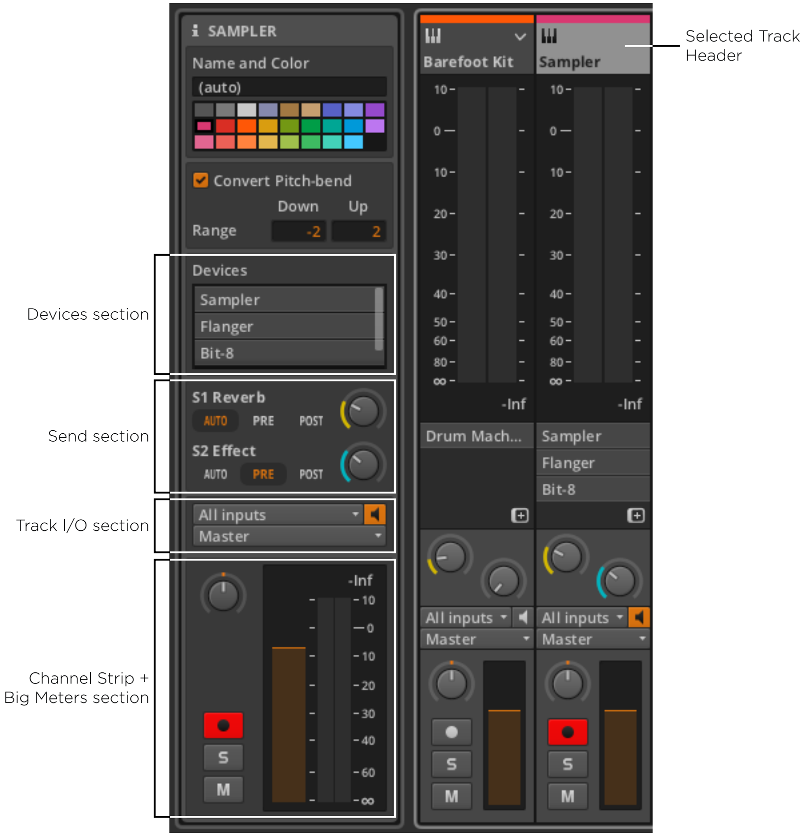
To resize a panel: mouse over the panel's border that faces the middle of the Bitwig Studio window. When the cursor becomes a bidirectional arrow, click and drag the border.

Now all enabled sections are visible, each working the same as they did in the central **Mixer Panel**.

The only difference in this secondary version of the panel is that the **Clip Launcher Panel** and the big meters section are unavailable.

6.2.2. Mixing in the Inspector Panel

Finally, the **Inspector Panel** will also display certain mixing parameters whenever a track is selected. Whether in the **Arranger Timeline Panel** or the **Mixer Panel**, clicking on the track header will focus the **Inspector Panel** on that track.



Both the devices and track I/O sections work exactly as they did in the **Mixer Panel**.

The send section has used the extra space here to display both the targeted effect track's name and the source settings (*auto*, *pre*, or *post*) for each send.

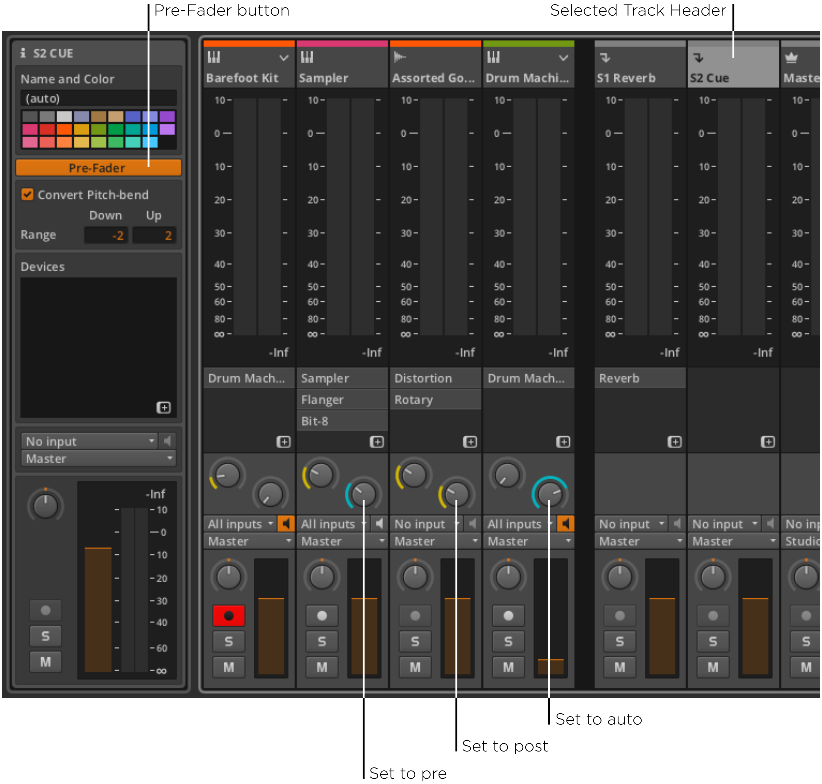
The *channel strip + big meters section* is a replica of the original channel strip section along with some rearranged big meters on the right. All functionality from these sections remains the same.

6.2.3. Inspecting Effect Tracks

Aside from lacking sends, everything just shown regarding the **Inspector Panel** holds true for effect tracks, but one additional parameter is worth noting.

To set a send track's routing preference: select the track and then toggle the *Pre-Fader* button from the **Inspector Panel**.

When a track's send source is set to *auto*, the destination effect track's routing preference will be used. By default, the preference of each effect track is *Post-Fader* (as indicated by *Pre-Fader* being deselected). When *Pre-Fader* is enabled, all corresponding track sends that are set to *auto* will now adopt a setting of *pre*.



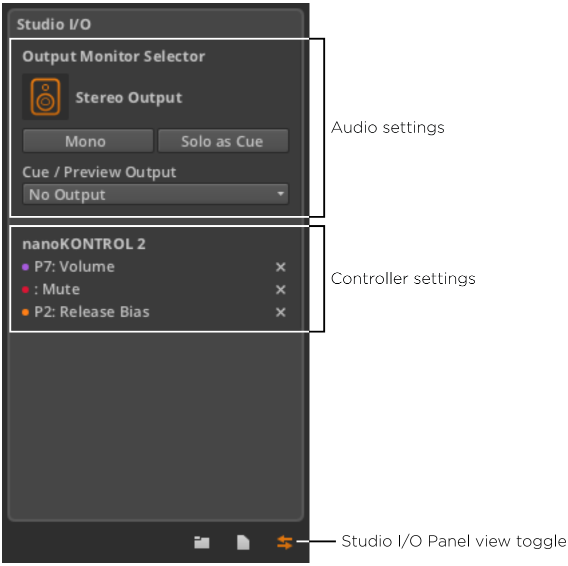
6.3. Master Track Routing

We mentioned earlier that the default output assignment of all tracks is *Master* (see section 4.4.1). This is referring to the name of the project's master track, which defaults to *Master*. If we rename the master track, the output choosers will follow suit.

As you can also see in the image above, the default output of the master track is set to *Studio*, which refers to the **Studio I/O Panel**. We will now examine this panel and then see an example setup where a multichannel audio interface is used.

6.3.1. Studio I/O Panel

Clicking the **Studio I/O Panel** view toggle in the window footer will call up the panel.



The top area of the panel displays the following audio settings:

› The *Output Monitor Selector* lets you select which pair of speakers and/or sets of headphones are being used for any track whose output is set to *Studio*.

  The monitoring options are those you have defined either on the first launch of Bitwig Studio (see section 0.1.2.2) or later in the Preferences window (see section 0.1.3) using the same interface.

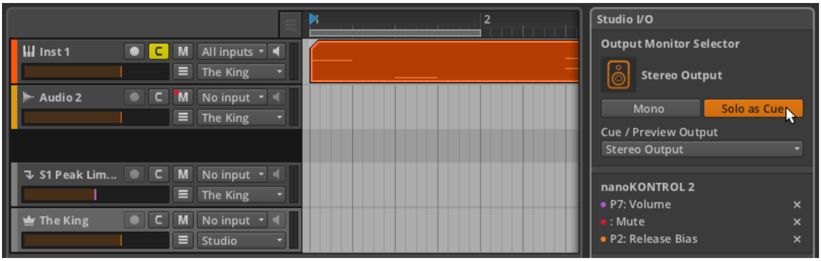
  To toggle a monitor: click the monitor's icon.

  Only one pair of speakers can be active at a time, and any number of headphones may be used.

  A fuller example with multiple monitoring options is presented in the next section.

› The *Mono* button toggles monitoring from stereo to a mixed-down mono output.

› The *Solo as Cue* button alters how solo works. When this function is enabled, all solo-enabled tracks are also routed to the cue output, and all other tracks are routed as usual. Solo buttons themselves will be switched from *S* to *C* to reflect this.



When this function is disabled, normal solo rules apply (see section 3.1.3).

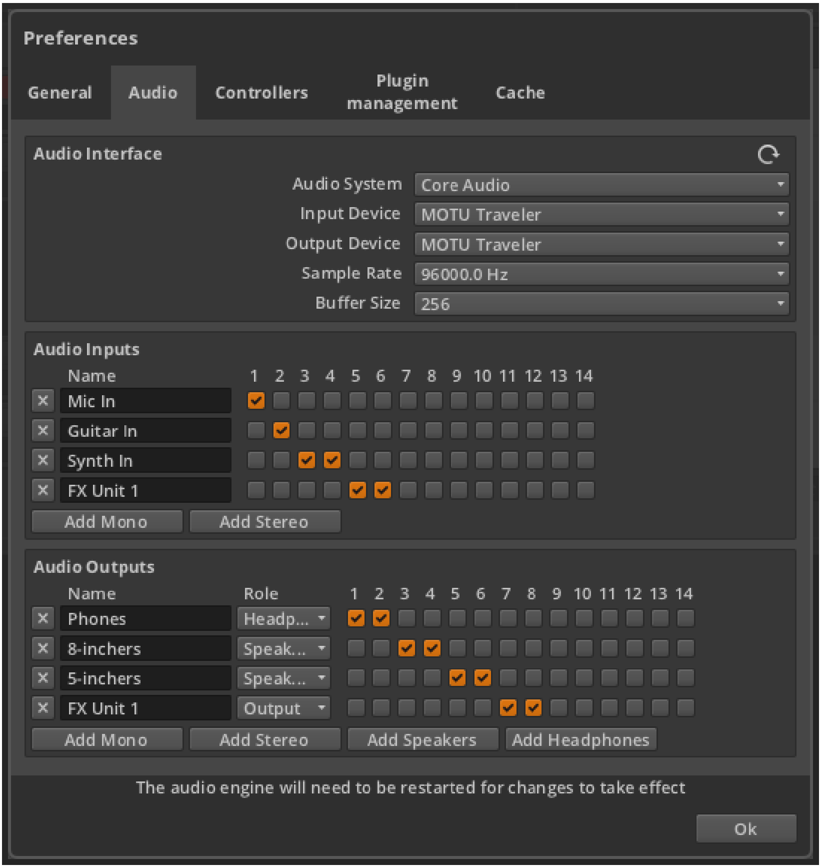
› The *Cue / Preview Output* menu sets the monitoring destination for both cue signals (when the *Solo as Cue* button is active) and **Browser Panel** previews.

This is particularly useful for performance situations. For example, this allows you to cue up certain signals in your headphones before adding them to the main mix.

The bottom area presents a list of all MIDI controllers currently connected or used in this project. Beneath each controller is a list of its manual controller assignments (see section 13.2).

6.3.2. Multichannel Audio Interface

Aside from the *Mono* button, all other audio settings in the **Studio I/O Panel** are only useful when you have more than one audio output option. To show one use case, I have connected a multichannel audio interface and made the following configuration in the *Audio* tab of the *Preferences* window.



Let's walk through the example shown above.

Under *Audio Inputs*, three paths have been set up:

› *Mic In* is a mono input path that uses input 1 of our audio interface.

› *Guitar In* is a mono input path that uses input 2.

› *Synth In* is a stereo input path that uses inputs 3 and 4.

› *FX Unit 1* is a stereo input path (for a hardware effects unit) that uses inputs 5 and 6.

Under *Audio Outputs*, four paths have been set up:

› *Phones* is a stereo output path that uses outputs 1 and 2 of our audio interface. This path has a role of *Headphones*.

› *8-inchers* is a stereo output path (for my 8" speakers) that uses outputs 3 and 4. This path has a role of *Speakers*.

› *5-inchers* is a stereo output path (for my 5" speakers) that uses outputs 5 and 6. This path has a role of *Speakers*.

› *FX Unit 1* is a stereo output path (for a hardware effects unit) that uses outputs 7 and 8. This path has a role of *Speakers*.

The audio input paths will now be available in various places in the program, such as audio tracks' input choosers.

The audio output paths will be available from every track's output chooser, but they will also appear in the **Studio I/O Panel**.

You will notice that only the monitoring options (*Speakers* and *Headphones*) are available here. Setting a path to an *Output* role makes it available for signal routing, but not for monitoring.

So in this example, my project's master track is routed to *Studio*. Because the *Output Monitor Selector* of the **Studio I/O Panel** is set to

8-inchers, anything reaching the master track is getting passed to my 8-inch speakers. And because *Solo as Cue* is enabled, any track that is solo-enabled (and any **Browser Panel** content being previewed) is routed to *Phones*.

If you have a simple setup and never click any of these options, audio will be routed to the right place. But if you have more sophisticated requirements, the settings shown here and Bitwig Studio's routing options will cater to your needs as well.

7. Introduction to Devices

The word "devices" has come up a few times now. For one thing, we have been using them already on instrument tracks (see section 4.4.2.1). For another, we have seen how other Bitwig Studio interfaces give us access to devices we were already using (see section 6.1.4). But in this chapter, we are finally dealing with the nuts and bolts of loading and using devices. This small exploration will benefit users of all levels.

Note

More "advanced" device concepts are covered in chapter 14, which assumes familiarity with the concepts found in this chapter.

The purpose of this chapter is not to teach you the particulars of any device. Instead, it is to acquaint you with accessing devices, their general interface concepts, and the layout of the **Device Panel**. A short section about the Bitwig devices themselves can be found at the end of this document (see chapter 15).

To expand slightly on chapter 1, each track in Bitwig Studio is equipped with a *device chain*. Each track passes all played-back audio, note, and MIDI signals to this device chain, which passes the messages from one device to the next, like a bucket brigade. The final device in the chain returns its audio output back to the track so that the mixing board controls (volume, panning, etc.) can be applied before the audio is passed to the track's assigned output buss.

Devices are grouped into the following functional categories:

› *Note FX* manipulate incoming note messages before passing them onward.

  Examples include **Arpeggiator** and various note transposers.

› *Instruments* are devices that use incoming note messages to synthesize audio.

  Examples include **Polysynth**, **Sampler**, and various electronic drum emulators.

› *Containers* are utility devices whose primarily function is to host other devices.

  Examples include **Drum Machine** (for individual note splits), **Instrument Layer** (for stacks), and **Multiband FX-2** (for multiband audio processing).

› *Audio FX* manipulate incoming audio signals before passing them onward.

Examples include various equalizers, **Compressor** and other dynamics-based effects, **Reverb** and other time-based effects, and all manner of modulation tools.

› *Generators* are devices that produce output without the need for any input.

› *Modulators* are devices that can modulate parameters within their own *FX* chain. Modulators are primarily used to manipulate other devices, but they can also influence some of their own parameters.

Examples include LFO and step sequencer-based modulators, as well as **Audio MOD** (a utility "sidechain" effect).

› *Routers* divert a track's signal path, allowing signals to exit and/or reenter the track. This includes the ability to route signals outside of Bitwig Studio itself.

Examples include **Hardware Instrument** (for routing MIDI out of Bitwig Studio and the resultant audio back in) and **Hardware FX** (for sending one stereo pair of audio signals out of the track and system, and then bringing another pair back).

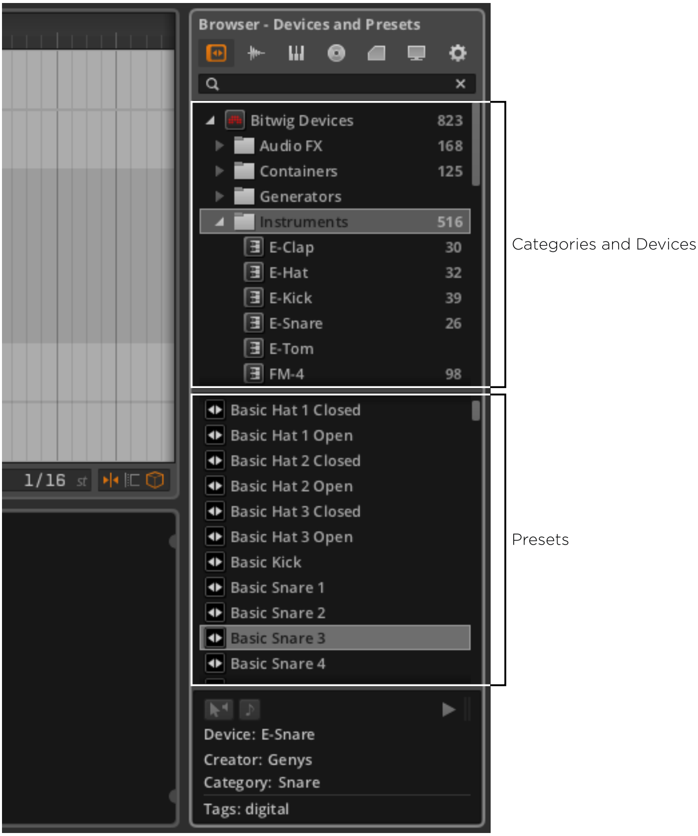
So while devices aren't always necessary, they can make things a whole lot more interesting and open up possibilities that you may not have previously thought of.

7.1. Accessing Devices

There are three ways to load devices into your Bitwig Studio project: by loading a preset from the **Browser Panel**, by loading a device from the **Browser Panel**, or by loading a device from the Add Device window.

7.1.1. Presets from the Browser Panel

The Devices and Presets tab of the **Browser Panel** is the only direct route to presets. This is a good way to browse for sounds themselves.



To recap from chapter 4 (see section 4.1.1), the *Devices and Presets* tab uses the navigation pane to display both the device categories and the devices themselves. Once a selection is made in that top pane, the selection pane below displays all corresponding presets.

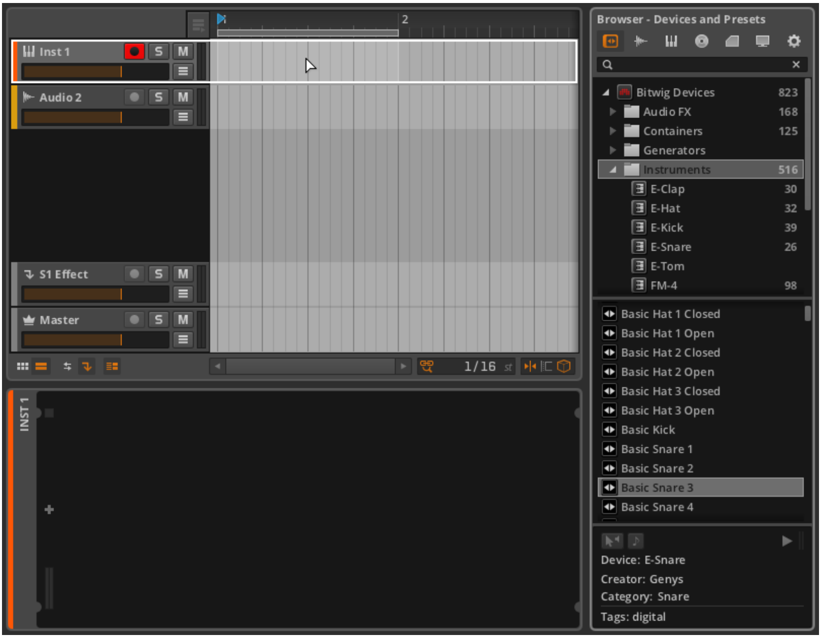
For example, after selecting the *Instruments* category, the selection pane shows all the standard presets that use a basic instrument device. This is the simplest way to browse for a particular sound.

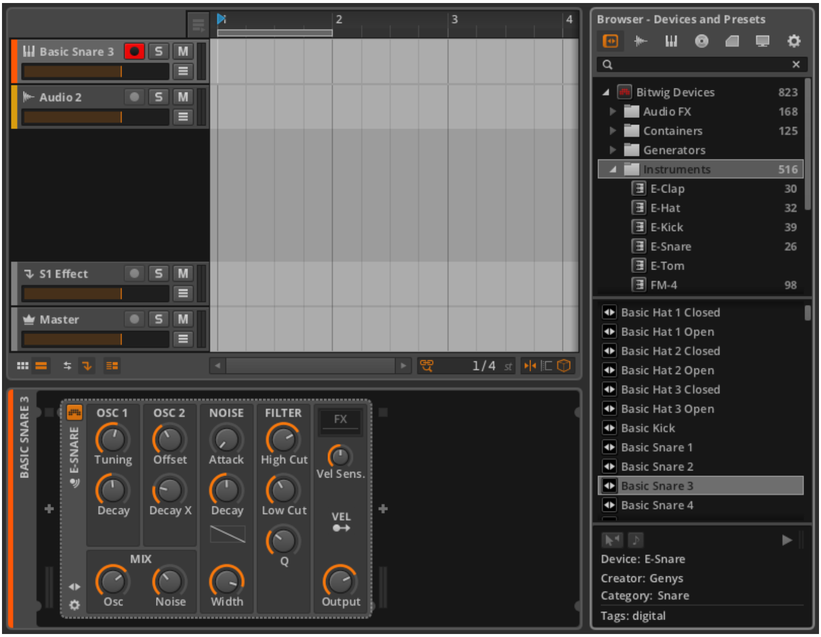
Note

The *Containers* category also has a number of instrument-driven presets. You can browse them by selecting *Containers* in the navigation pane and scrolling through the selection pane for any preset with a white icon.

You could also select presets from the top-level *Bitwig Devices* category itself, but this will include presets from all device types. Targeted searching is most easily accomplished by browsing presets within specific device categories.

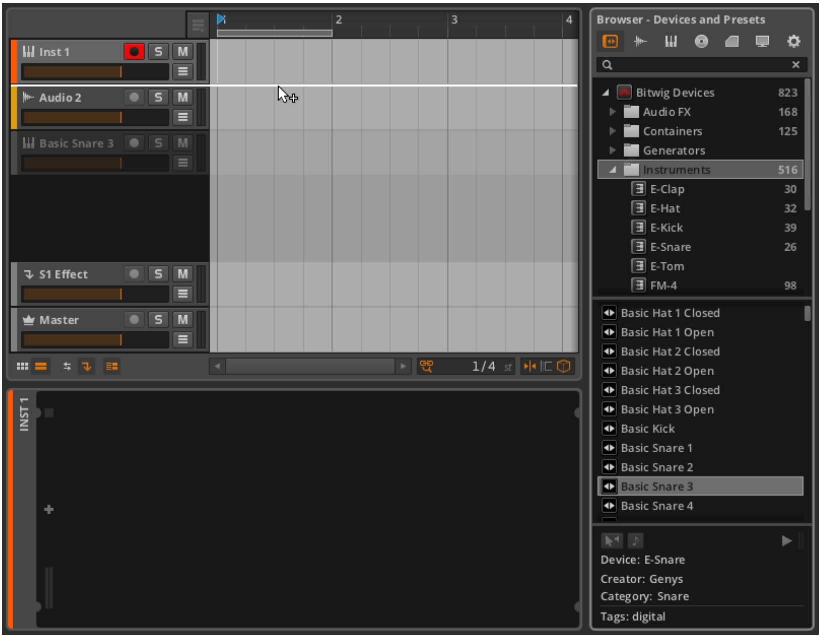
To insert a preset and device onto an existing track: click and drag the preset from the **Browser Panel** to the appropriate track.





This can be achieved in any of the editing panels, including the device section of the **Inspector Panel** when the target track is selected.

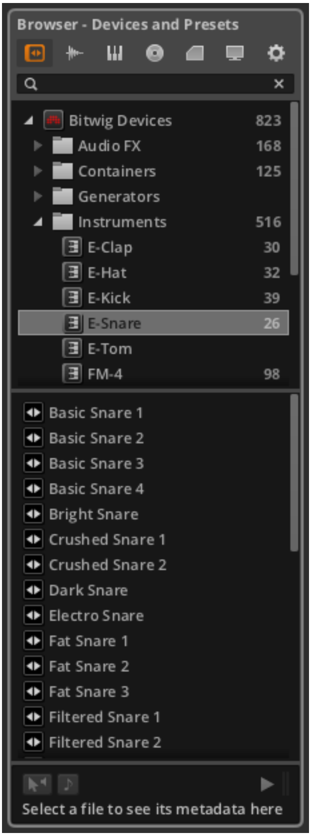
To insert a preset and device onto a new track: click and drag the preset from the **Browser Panel** to the space between existing tracks.



This requires either the **Arranger Timeline Panel** or the **Mixer Panel**, as these panels are made to display the entire project at one time.

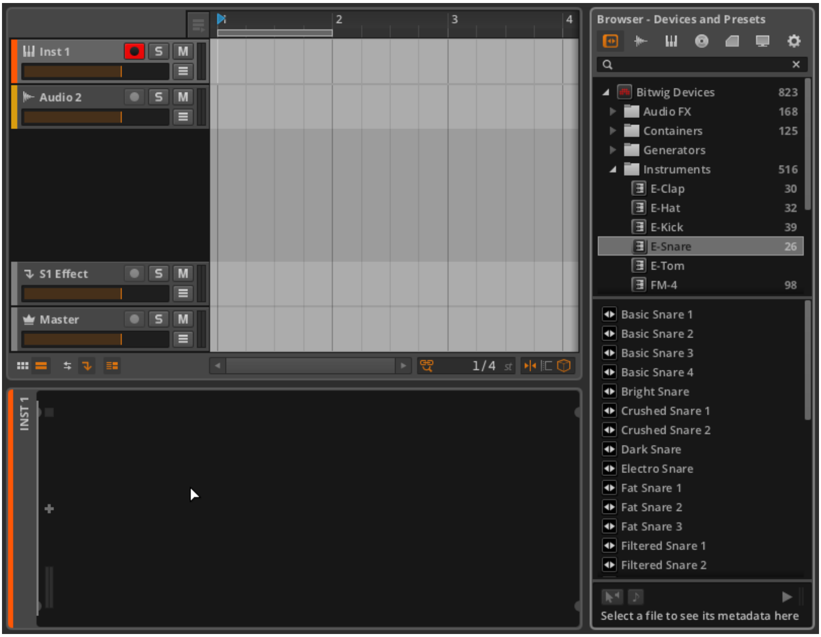
7.1.2. Devices from the Browser Panel

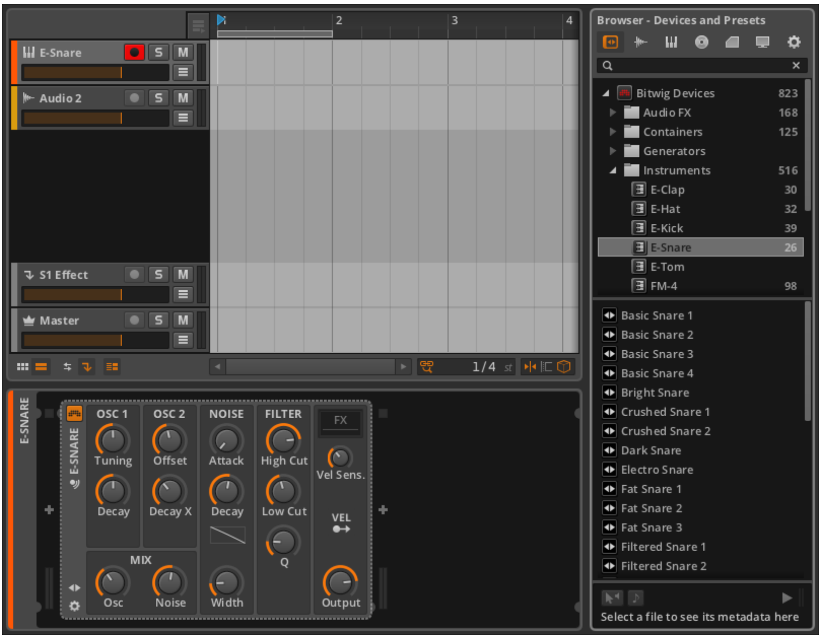
Using the browser again, you can select a particular device to work with.



Now that a single device is selected, only presets made with that device are listed in the selection pane. You can now either import one of the selected device's presets as described a moment ago, or you can import the device itself with its default settings.

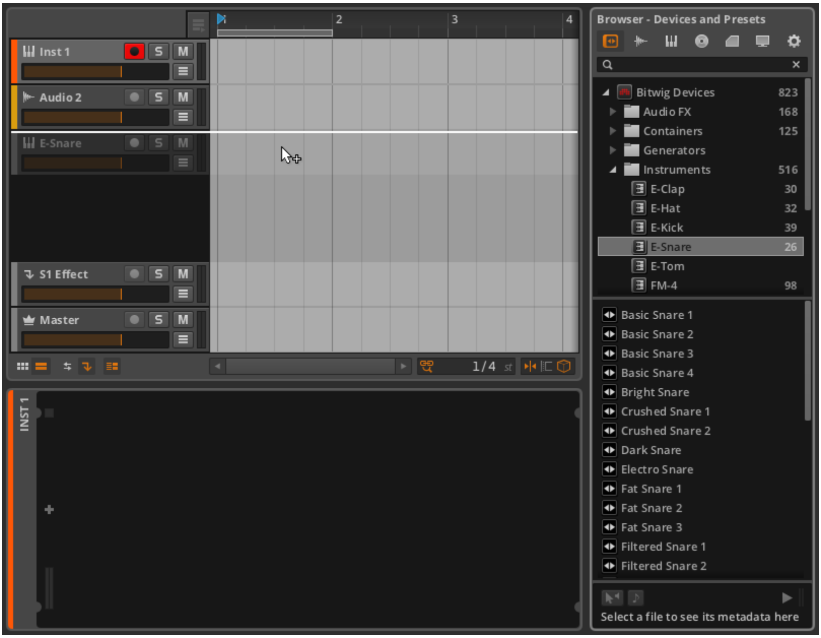
To insert a device onto an existing track: click and drag the device from the top of the **Browser Panel** to the appropriate track.


Inst 1
Audio 2
S1 Effect
Master
INST 1
1/4
Browser - Devices and Presets
Bitwig Devices 823
Audio FX 168
Containers 125
Generators
Instruments 516
E-Clap 30
E-Hat 32
E-Kick 39
E-Snare 26
E-Tom
FM-4 98
Basic Snare 1
Basic Snare 2
Basic Snare 3
Basic Snare 4
Bright Snare
Crushed Snare 1
Crushed Snare 2
Dark Snare
Electro Snare
Fat Snare 1
Fat Snare 2
Fat Snare 3
Filtered Snare 1
Filtered Snare 2
Select a file to see its metadata here


E-Snare
Audio 2
S1 Effect
Master
E-SNARE
1/4
OSC 1
Tuning
Decay
OSC 2
Offset
Decay X
NOISE
Attack
Decay
Width
FILTER
High Cut
Low Cut
Q
FX
Vel Sens.
VEL
MIX
Osc
Noise
Output
Browser - Devices and Presets
Bitwig Devices 823
Audio FX 168
Containers 125
Generators
Instruments 516
E-Clap 30
E-Hat 32
E-Kick 39
E-Snare 26
E-Tom
FM-4 98
Basic Snare 1
Basic Snare 2
Basic Snare 3
Basic Snare 4
Bright Snare
Crushed Snare 1
Crushed Snare 2
Dark Snare
Electro Snare
Fat Snare 1
Fat Snare 2
Fat Snare 3
Filtered Snare 1
Filtered Snare 2
Select a file to see its metadata here

In this case, the device was dragged into the **Device Panel**, which was focused on the appropriate track.

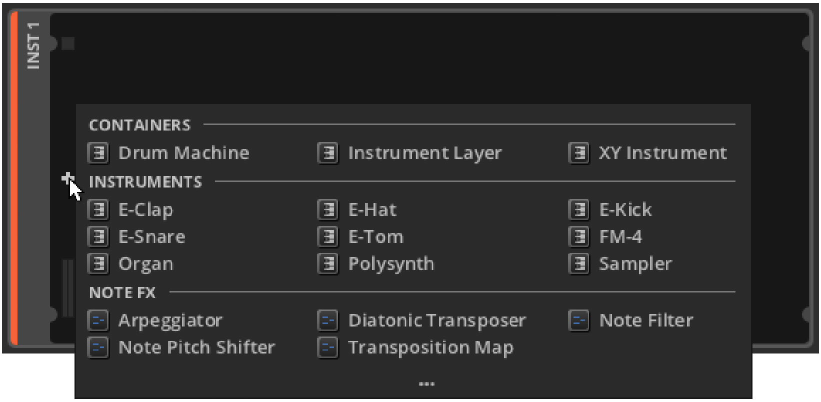
To insert a device onto a new track: click and drag the device from the **Browser Panel** to the space between existing tracks.



7.1.3. The Add Device Window

Outside of the **Browser Panel**, there is one other way to load devices. The *Add Device window* is a pop-up window that is available in a few different places throughout Bitwig Studio. We have already seen it in the device sections of both the **Mixer Panel** and the **Inspector Panel** when a track is selected.

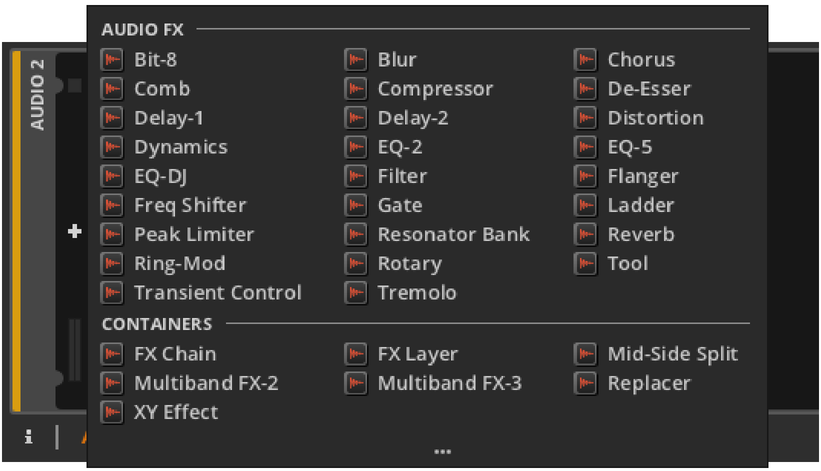
The Add Device window is also available in the **Device Panel** by clicking on the *Add Device button* (+).



This Add Device function is aware of its surroundings. It first assesses where it was called up within the track's signal flow, and it then gives you the most common choices for the situation.

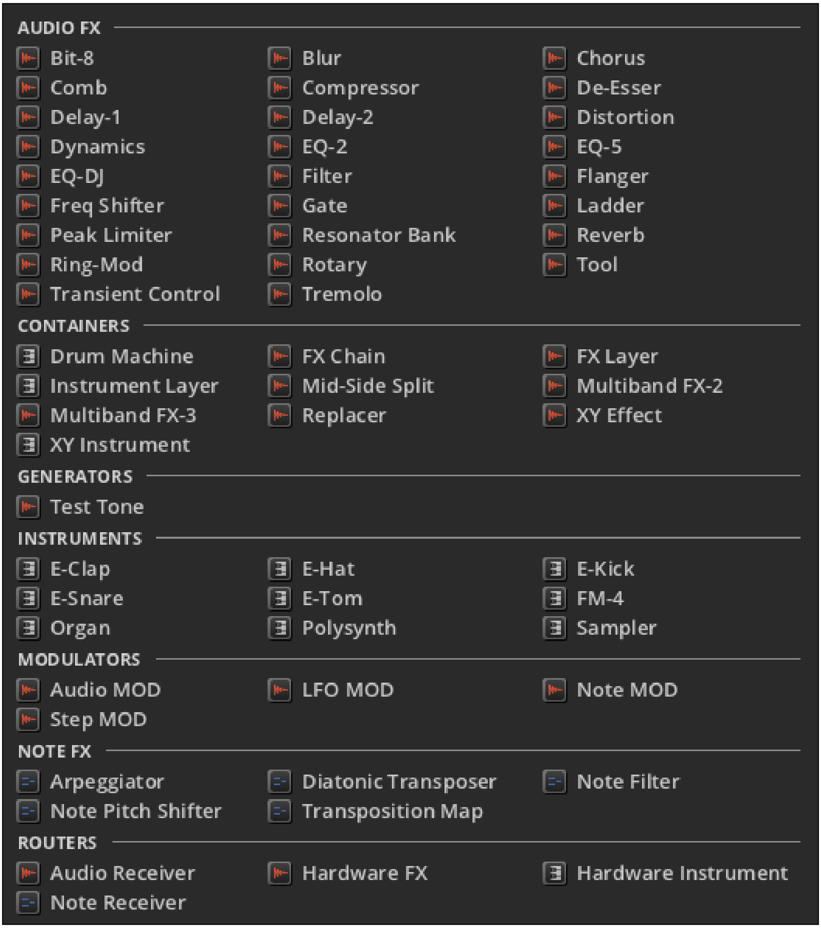
In the image above, the Add Device window was called up in a blank instrument track. In that situation, note messages are expected from the track so the Add Device window offered common devices that expect note input (note FX, instruments, and containers that take note input).

If we were to click the Add Device button on a blank audio track, the result would be different.



In this case, common devices that expect audio input are being offered (audio FX and containers that take audio input).

At the bottom of both windows is an ellipsis icon (...). Clicking this icon calls up the full Add Device window.



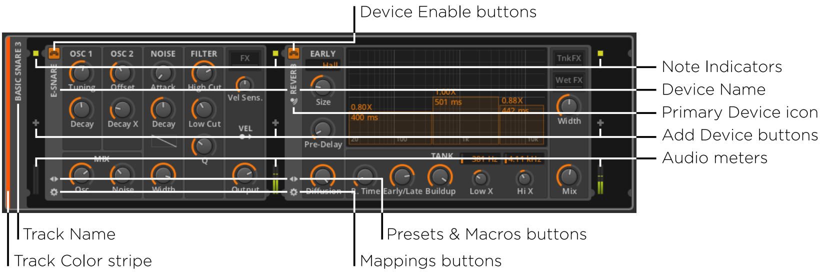
This version of the Add Device window simply lists all categories and devices, allowing you to make both common and uncommon choices.

7.2. The Device Panel

Whether we load our devices from other panels or not, the **Device Panel** is where all direct interaction with devices will occur. So once we are ready to work with devices, we must explore the **Device Panel** and see what it has to offer.

7.2.1. The Panel Itself

Let's take a simple example of a track that contains two devices: one instrument and one audio FX.



Note that the above image shows the instrument on the left and the audio FX on the right. In the **Device Panel**, signal always flows from left (input) to right (output). While you could swap the position of these devices, you probably would not get the desired outcome.

Starting with the outer rounded rectangle, we find on its left edge an abbreviated, vertical track header. Included here are the familiar *track color stripe* and *track name*.

Other than the track header, all space in the **Device Panel** is reserved for devices. But before the first device (and after every device) comes a vertical column containing three items:

› The *note indicators* light up when at least one note message is active at that stage. (This is similar to a MIDI "note on" message that has not yet been followed by a corresponding "note off.")

› The *Add Device button* calls up the Add Device window.

› The *audio meters* indicate the presence and level of audio signal being received and transmitted by each device.

The Add Device button is present in all these locations so that you can insert additional devices at any point within the device chain. The note indicators and audio meters are present at every device handoff to visually inform you of signals that are changing as the signal flow progresses. As relevant texts and your own experimentation will teach you, the order in which devices are connected is critical to the outcome.

Each device has its own vertical header on its left edge. Common elements in device headers are:

› *Device Enable button*: Toggles the device between on (engaged) or bypass mode (disabled).

› *Device Name*: The official name of the device.

› *Primary Device icon*: Indicates that the device is the primary focus on this track of attached MIDI controller hardware (see section 13.1.2).

› *Presets & Macros button*: Toggles to reveal the Presets & Macros pane for the device.

› *Mappings button*: Toggles to reveal the Mappings pane for this device (see section 13.1.2).

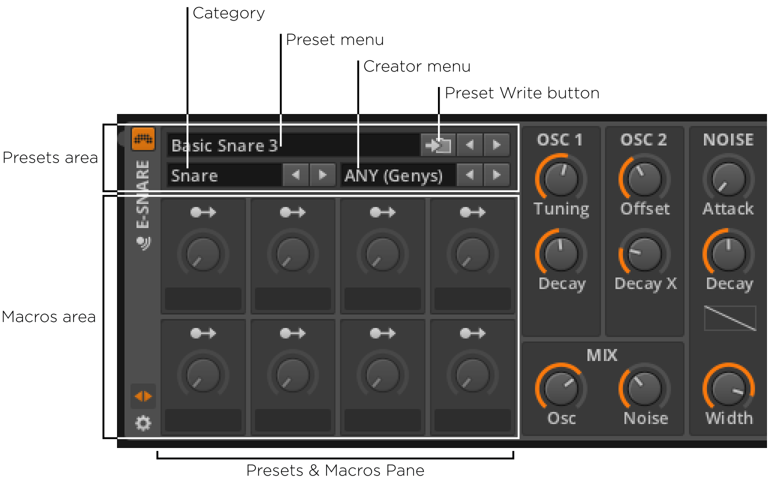
Finally, the body of each device contains its own various parameters. They can take the form of knobs, sliders, numerics, text and graphical lists, buttons, curve controls, clickable graphic interfaces, and more. All parameters can be set with the mouse by simply clicking and dragging.

7.2.2. Presets & Macros Pane

Right-clicking the Presets & Macros button reveals a list of presets for the device.

The list is organized into the available preset categories. By selecting a preset from the list, the current settings of the device are replaced with those of the preset.

If you single-click the Presets & Macros button, the *Presets & Macros pane* is toggled to be visible.

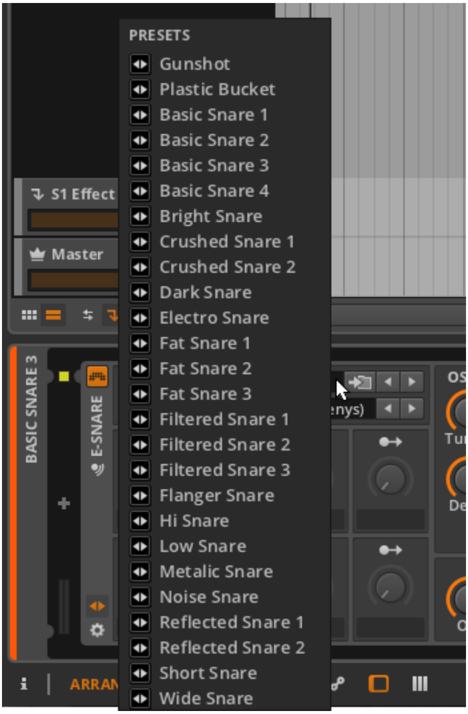


Note

The *macros area* of this pane is discussed in section 14.2.1.

The *presets area* of this pane is arranged to let you browse a device's presets in several ways. In this area are three menu options.

› *Preset menu*: A pop-up menu of all presets for this device that match the current category and creator selections.



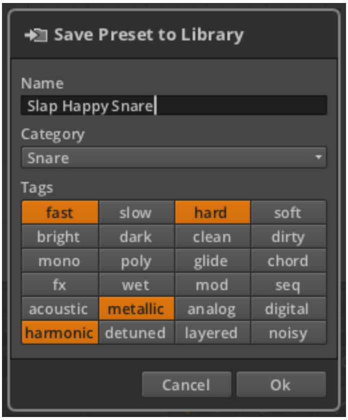
› *Category menu*: A pop-up menu of every preset category used for this device. The setting here is used to constrain the preset menu. The default setting is *ANY*.

› *Creator menu*: A pop-up menu of every preset creator listed for this device. The setting here is used to constrain the preset menu. The default setting is *ANY*.

To the right of each of these menus is a pair of incrementer/ decrementer buttons, which allow you to browse forward and backward through the respective list.

The *Preset Write button* is a critical addition that allows you to store your own custom presets.

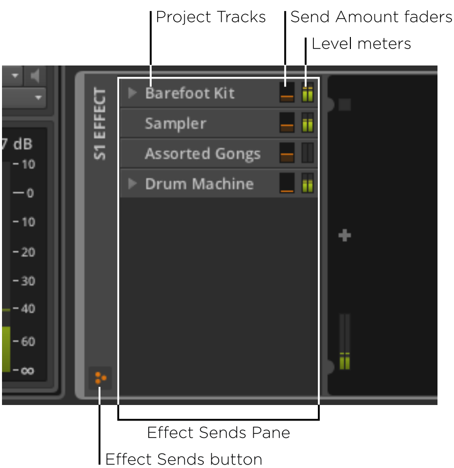
To store a device's current settings into a preset: click the Preset Write button. In the *Preset Write window* that appears, set the new preset's name, assign the proper *Category*, set any appropriate *Tags*, and select *Ok*.



Assigning proper settings whenever you save a preset will make finding them much, much easier later.

7.2.3. Effect Tracks and Send Amounts

Effect tracks have one unique feature in the track header of the **Device Panel**.

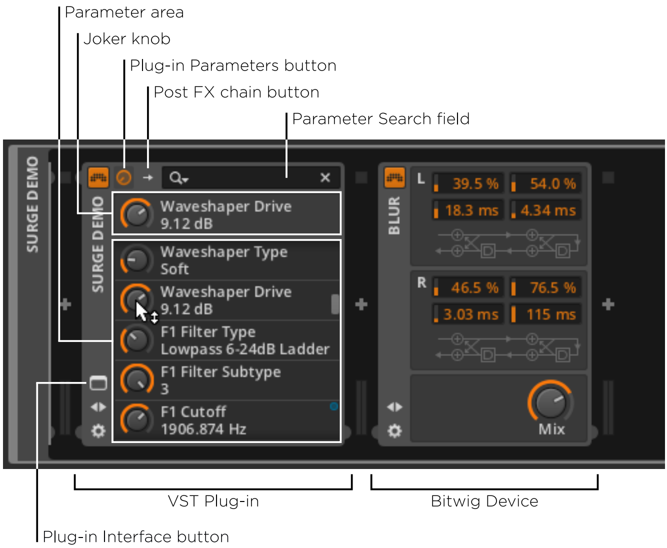


When the *effects sends button* is enabled, the *effect sends pane* is visible within the track header area. This resizable pane shows a list of all instrument, audio, and hybrid tracks in your current project. Each track is listed along with a meter showing its current output level and a control for the send amount targeting this effect track.

Essentially, this is a "mixer" view of the buss that feeds the effect track. And tracks that have track fold buttons on the mixer (see section 6.1.1) have a similar fold button here.

7.3. VST Plug-ins

The other kind of device that can be used in the **Device Panel** is VST plug-ins. Aside from setting up Bitwig Studio to recognize the plug-ins you own (see section 4.1.7), we haven't talked much about them. They operate side by side with Bitwig devices, and both generally function in the same way, but the interface for plug-ins is a bit different.



The bulk of the panel is reserved for the VST's *Parameter area*, but the parameters are in the form of a long scrollable list of knobs. And above this list is a single *joker knob*, which is really an alias (or wildcard) which follows the last plug-in parameter that you touched. So after you scroll halfway down a very long parameter list, the last parameter you adjusted will still be accessible just above the list.

The top row of most plug-in devices has three important controls:

› The *Plug-in Parameters button* (with a knob icon) is lit up whenever the joker knob and list of parameters are being shown below.

› The next button varies depending on the type of plug-in your have loaded:

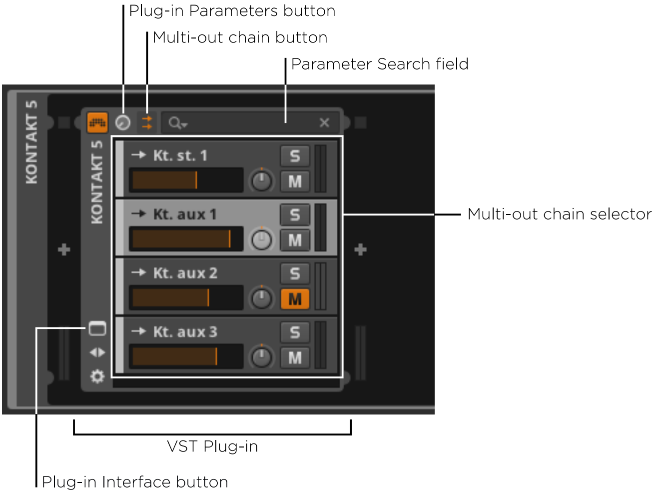
Most plug-ins then have a *Post FX chain button* (with a single right-facing arrow for an icon), as was shown in the above image. Clicking this button expands the right edge of the plug-in interface to display a chain where other devices and plug-ins can be loaded.

The advantage of loading devices within a plug-in's Post FX chain is that when you store a preset for this plug-in, that preset will include all attached devices as well as their settings. So in the example above, saving a preset for **Surge** would include the **Chorus** device and the **MasterVerb 5** plug-in along with all of their current settings, but the **Blur** device would not be included.

Note

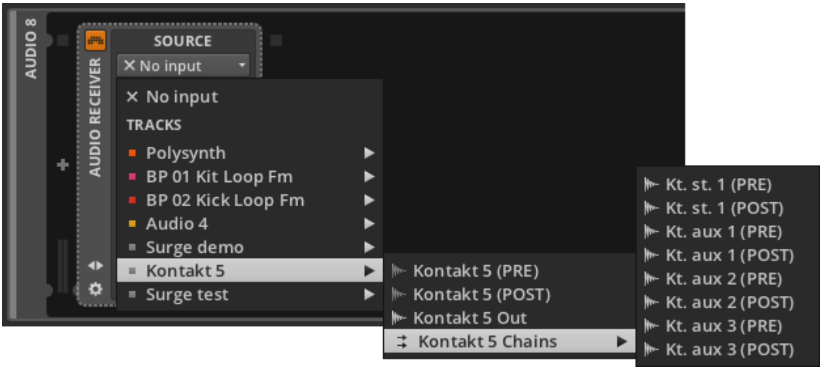
For more information on nested device chains, see section 14.1. And for specific information on Post FX chains, see section 14.1.3.

Multichannel plug-ins do not have a Post FX chain button and corresponding chain. Instead, they have a *Multi-out chain button* (with two right-facing arrows for an icon). Clicking this button toggles the parameter area below to instead show the *Multi-out chain selector*.



This chain mixer gives you mixing controls for all the various outputs of this multichannel plug-in within the current stereo track. Clicking on the plug-in parameters button will return the parameter area to its normal joker knob and parameter list.

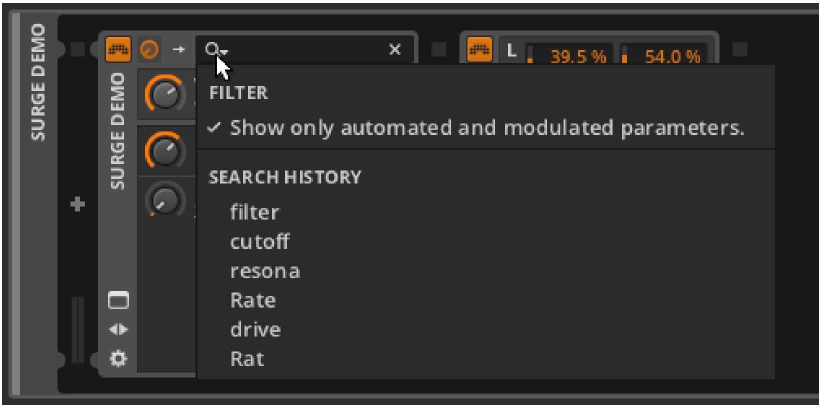
To access audio channels from a multichannel plug-in on a different track: either from a track's audio input chooser or from an **Audio Receiver** device's *SOURCE* menu, select the track of the multichannel plug-in, then select its *Chains* submenu, and finally select the desired audio source.



› The *Parameter Search field* is provided in the top of the plug-in area to let you filter the parameter list and find what you are looking for.

This is useful as the parameter list for a complex plug-in can be exceedingly long.

The parameter search field's magnifying glass icon also doubles as a menu. By clicking on this icon, you can thin the parameter list to *Show only automated and modulated parameters.* You can also revisit your recent *SEARCH HISTORY* from this menu.



Finally, in the device header is now a *Plug-in Interface button*. Clicking this button calls up the plug-in's custom interface in a floating window.

(As all plug-ins have their own custom interface, please don't expect anything else to look like **Surge**, shown as the example above.)

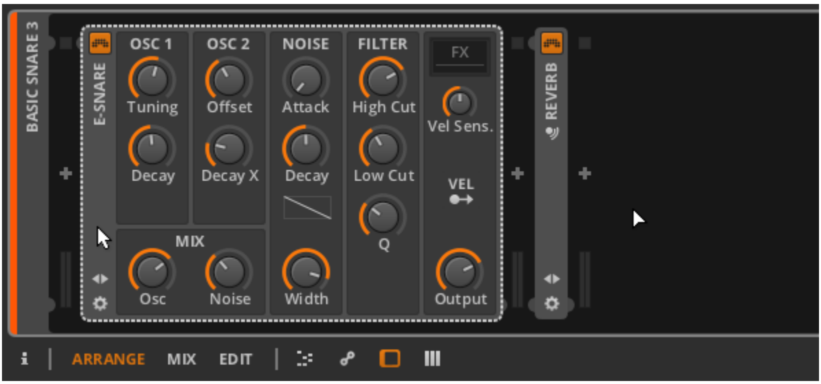
7.4. Working with Devices

Earlier in this chapter, we covered both adding devices and loading presets. Before moving on, here is a list of other basic functions you may wish to execute with the **Device Panel**.

To minimize/restore a device's interface: double-click the device header.

This is a change in appearance only and does not affect the operation of any device.

To select a device: single-click its track header.

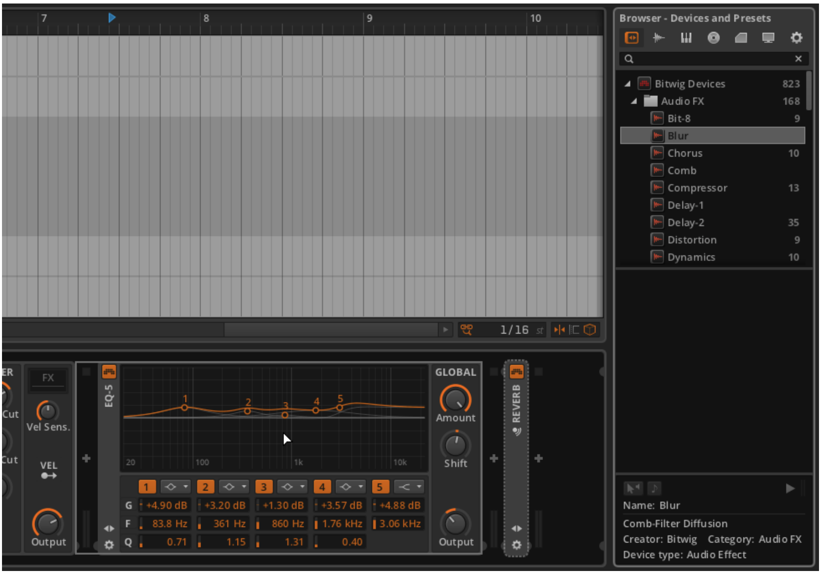


The currently selected device is indicated with a dashed white border. Once selected, all regular *Edit* functions apply, such as cut, copy, duplicate, and delete.

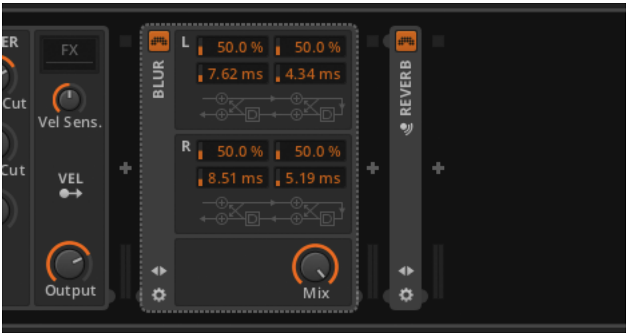
To move devices around: click and drag the device header to the desired position within the **Device Panel**.

As the status message in the footer suggests, [CTRL] ([ALT] on Mac) can be added to toggle the move to a copy function.

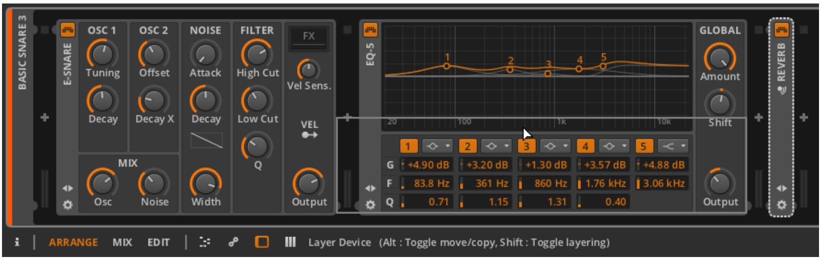
To replace one device with another: drag the desired device or preset from the **Browser Panel** onto the device to be replaced.



Once the mouse is released, the device will be replaced.



To layer a device with another: [SHIFT]-click and drag the device over top of the device where the layer should be inserted.



Depending on the type of devices being layered, an appropriate container device will be created and populated with your selections.

Note

For more information on container devices and other advanced device concepts, see chapter 14.

8. Automation

With the mixer interface (chapter 6) and our introduction to devices (chapter 7), we examined both track and device parameters that you will want to set as your own tastes dictate. But fixing these parameters to certain values is probably not enough.

If you can think about how a song develops — from the arrangement growing as parts gradually fade in and find their place in the stereo field, to instruments becoming more animated as their tones morph and brighten, to parts gradually fading away by both losing volume and increasing reverb — then you can visualize the series of long and short curves that represents a piece of music and its structure.

Automation is the animation of any defined parameter over time. It is usually thought of as narrative and rigid (in the same way the Arranger Timeline defines an exact musical progression), but Bitwig Studio also supports both a clip-oriented approach to automation, and techniques for having multiple layers of control cooperate to shape individual parameters in a relative way.

We will start our look at modulation back in the **Arranger Timeline Panel**, where we can work directly with traditional track-based automation. Then we will meet the **Automation Editor Panel**, whose sole purpose is displaying and manipulating automation. Finally, we will see how clip-based and relative automation can enhance our workflows and music in ways both novel and powerful.

Let's get those parameters dancing.

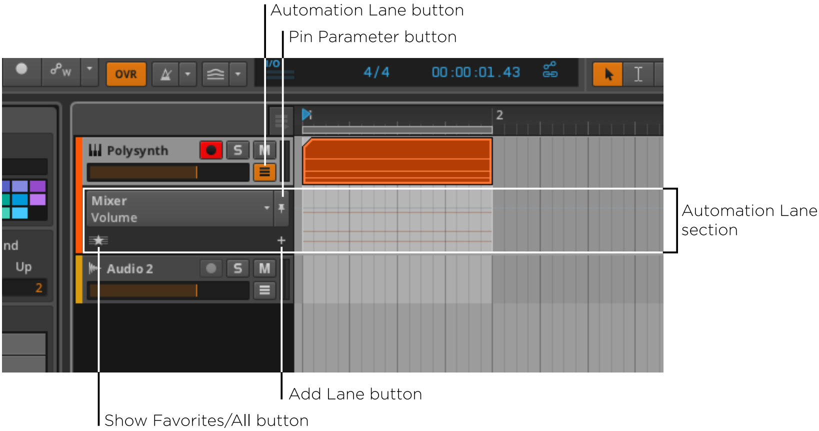
8.1. Automation Basics

If you work with music software and are used to only one type of automation, it is *track automation*. With this kind of automation, values for a parameter — volume, cutoff frequency, reverb amount, etc. — are stored as fixed values. So when the playback head reaches an automation point of either *-9.43 dB* or *2.88 kHz* or *124 %*, that exact value will be set and preserved until the automation dictates otherwise.

Bitwig Studio can accommodate this kind of automation, and it can be achieved with our old friend, the **Arranger Timeline Panel**.

8.1.1. The Arranger's Automation Lane Section

The one item in the Arranger that we have not looked into yet is the *Automation Lane button* within each track header. When a track has this button enabled, the *Automation Lane section* for that track becomes visible.



The Automation Lane section appears just below the track header and extends across the Arranger Timeline area as a place to show its own time-based data. Like all automation lanes, this one is resizable.

This track header section contains the following controls:

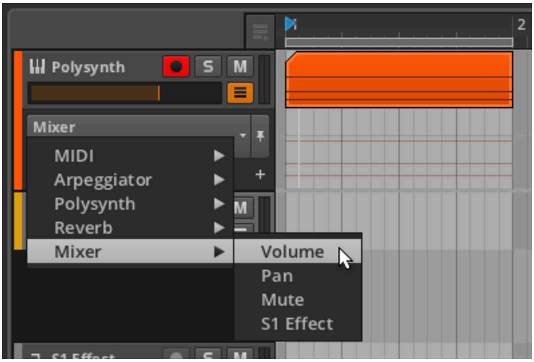
› *Parameter chooser*: Indicates and selects which parameter is displayed in this primary lane.

› *Pin Parameter button*: Maintains this lane's focus on the current parameter. This is disabled by default, which causes focus to follow the last clicked parameter.

› *Add Lane button*: Creates an additional automation lane that is fixed on the currently selected parameter.

› *Show Favorites/All button*: Toggles between displaying additional lanes for either your favorite parameters or for all parameters that are automated.

By clicking on the Parameter chooser, we will see a list of all automation targets for the selected track.

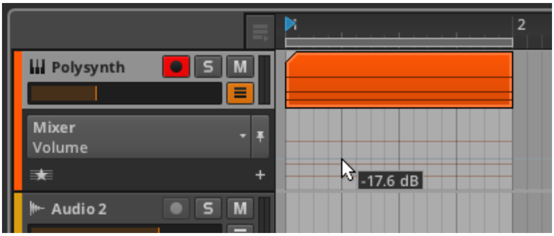


The top-level of the list always starts with various *MIDI* messages that can be stored on this track, and it always ends with the track's *Mixer* elements. In between these sub-menus will be a menu for each device preset on this track.

The background of the Automation Lane in the Arranger Timeline faintly hints at notes or audio events on the current track. These cannot be selected or edited; they are just illustrations to help you define your automation in relation to the track's contents.

This area is where our automation functions will be defined. And while this lane might seem empty, one subtle datum is present.

As the picture above shows, there is a light gray line just above the note outlines. This is the current automation curve of the track's volume. And since there are no additional automation points, that curve is a flat line at the current setting of *+0.00 dB*. If we were to grab the volume fader in the track header and make it quieter (by dragging it to the left), the gray line would follow.

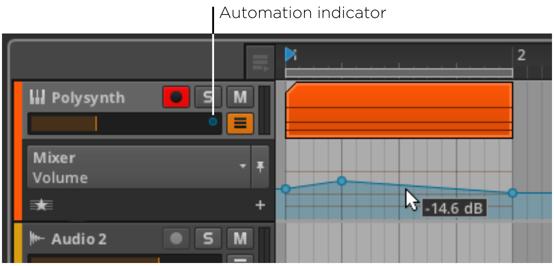


8.1.2. Drawing and Editing Automation

We will start with manipulating single points of automation. Similar processes will also work when multiple values are selected.

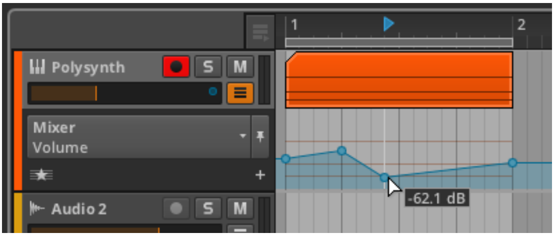
To create a single point along the automation curve: click in an area along the curve, and then drag the point to the desired value and position. Or single-click anywhere within the automation lane with the Knife tool.

We can repeat this a few times to create a small shape.



Note that dragging your mouse along the automation curve displays the parameter value beside your cursor for that song position. Also note the blue circle that has appeared near the top of the volume fader's range. This *automation indicator* — which looks like a misplaced automation point — indicates that the parameter in question is under the control of automation.

To create a single point outside the automation curve: double-click any area of the automation lane.

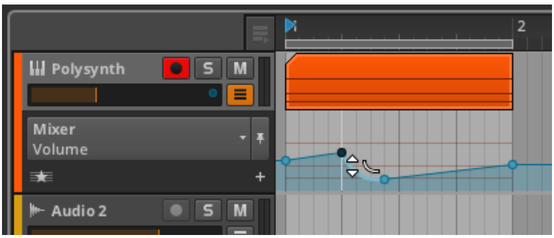


To move an automation point: click and drag the point with the mouse.

Note

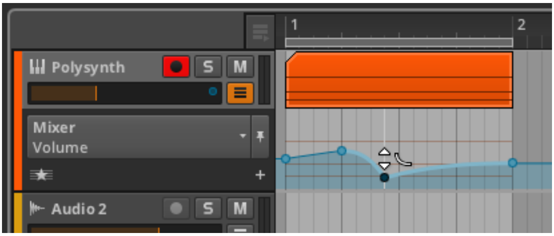
The absolute grid setting constrains the creation and movement of automation points. To temporarily toggle this setting, hold [SHIFT] while placing points.

To adjust the transition between two automation points: [ALT]-click and drag the curve between two points.



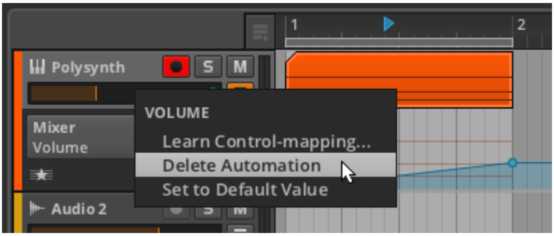
To reset a transition (to linear interpolation): [ALT]-double-click the transition.

To shape both transitions around an automation point: [ALT]-click and drag the point.



To delete an automation point: double-click it. Or single-click the point to select it, and then press [DELETE] or [BACKSPACE].

To delete all automation for a parameter: right-click on the parameter and select *Delete Automation* from the parameter's context menu.



To redraw an automation curve: click and drag horizontally with the Pen tool.

Once you release the mouse, the curve will be optimized to maintain its shape with the minimum number of points.

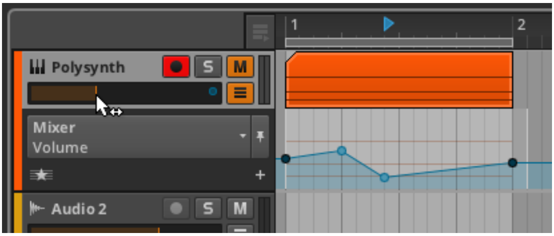
To select multiple points: click and drag a selection rectangle around the points of interest .

8.1.3. Parameter Follow and Automation Control

While you could use the Parameter chooser every time you need to find a parameter, the chooser can help you. Its default behavior is to focus on whichever parameter you select with the mouse. We call this initial automation lane the *joker lane* because like a wild card, it takes on whatever function you want it to.

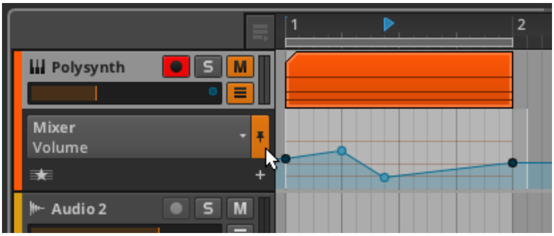
For example, clicking the track's mute button will now focus on the primary lane for that parameter.

If you then click on the track's volume fader, focus will return to the volume parameter.



As you can see, the automation that was drawn a minute ago has not been lost. This primary lane is simply shifting its focus with each mouse click.

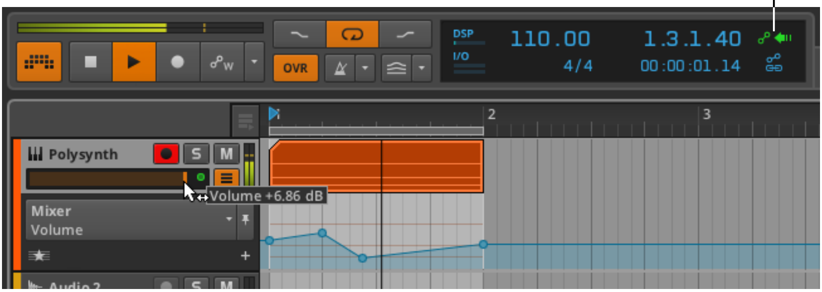
To lock the Parameter chooser to its current selection: enable the Pin Parameter button.



In the example shown, the Parameter chooser will now stay focused on the *Volume* parameter even if you click on the track mute button or any other parameter.

Additionally, Bitwig Studio will let you temporarily override the automation values you have set. This will automatically occur whenever you grab an automated parameter and adjust it.

Restore Automation button

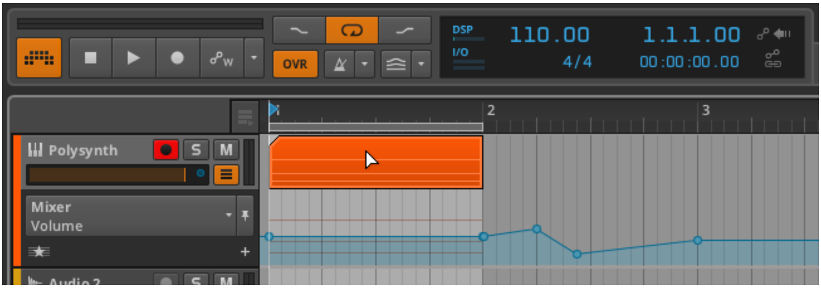


The automation indicator for the volume parameter has switched from blue to green, indicating that the automation's control of this parameter has been broken for the time being. At the same time, the *Restore Automation Control button* within the display section of the window header is now tinted green, indicating that it is armed.

To restore the control of automation over all parameters: click the Restore Automation Control button.

The *Automation Follow button*, which is also in the window header's display section, is worth mentioning here. This button toggles whether track automation is moved in tandem with Arranger clips or not. The setting is enabled by default so moving a clip would have the following effect.

Disabling the button and moving the clip back would leave any and all track automation behind.

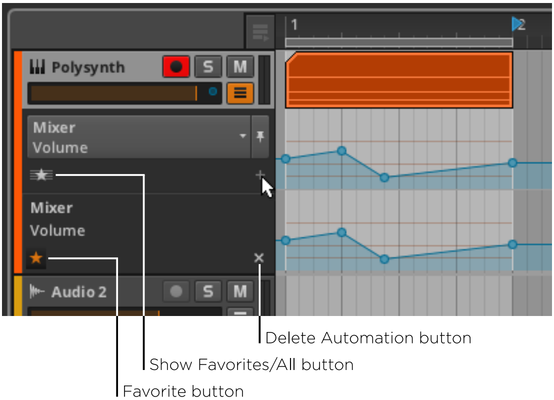


The would hold true for movement functions, such as copy, duplicate, etc.

8.1.4. Additional Automation Lanes

At times it will be useful to view the automation curves of several parameters at once. To achieve this, Bitwig Studio supports fixed automation lanes that appear beneath the dynamic primary lane.

To create a fixed automation lane for a parameter: select the desired parameter in the chooser, and then click the *Add Lane button*.



While it looks as though the lane just duplicated itself, there are some key differences here.

Only the top lane has a parameter chooser. The new lane — and any subsequent lanes — only has a text label indicating the device and parameter being automated so it cannot change focus.

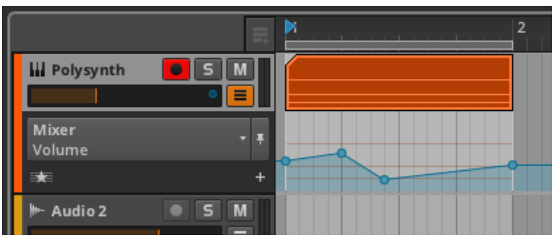
You will also notice that the new lane has two slightly different interface buttons beneath.

› *Favorite button*: Marks the parameter to be displayed in the favorites list.

› *Delete Automation button*: Deletes all automation for the lane's parameter and removes the lane.

As the *Show Favorites/All button* above is indicating with its star icon, tracks default to displaying favorite parameters. When favorites are being shown, clicking the Add Lane button both creates the fixed lane and automatically marks this parameter a favorite. The enabled Favorite button of our new lane demonstrates its status.

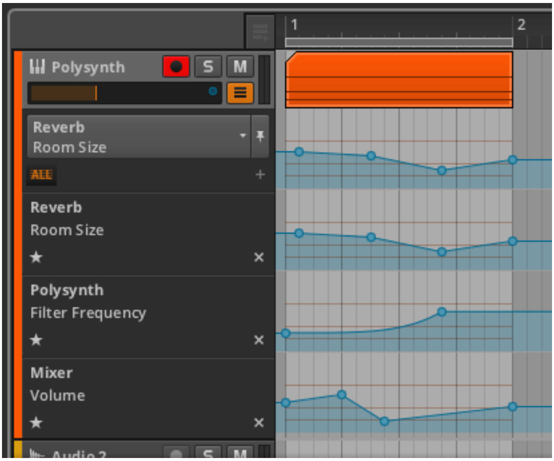
To remove a fixed lane's favorite status: disable the lane's Favorite button.



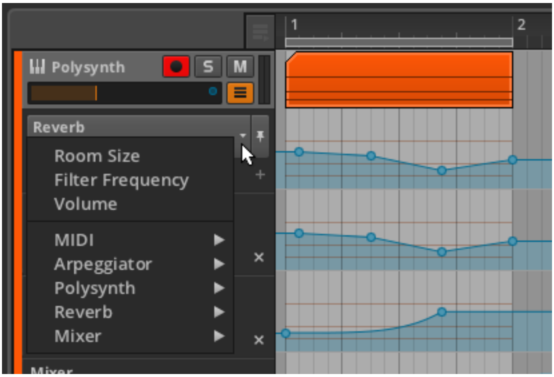
This essentially puts us back to where we started.

Please do not confuse the *Delete Automation button* for a "close" button. Clicking it instead of the Favorite button will collapse the additional lane, but it will also delete the automation for that parameter.

To display all parameters that have automation: toggle the Show Favorites/All button to the *All* setting and icon.

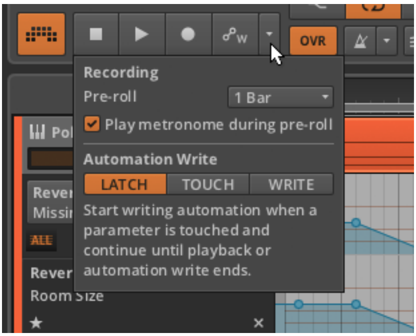


The list of automated parameters can also be accessed from the top of the Parameter chooser list.



8.1.5. Recording Automation

The automation write mode is set by clicking on the *Automation Write mode* menu in the window header's transport controls section.



There are three modes for recording automation.

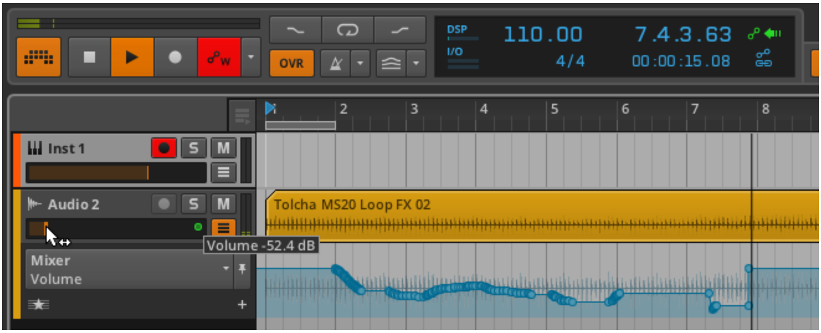
› *Latch* mode begins recording automation values as soon as you begin changing parameters. Recording then continues until the transport is stopped.

› *Touch* mode also waits until you have begun changing parameters to begin recording automation values, but once you stop interacting with a parameter, recording is halted and any preexisting values are preserved.

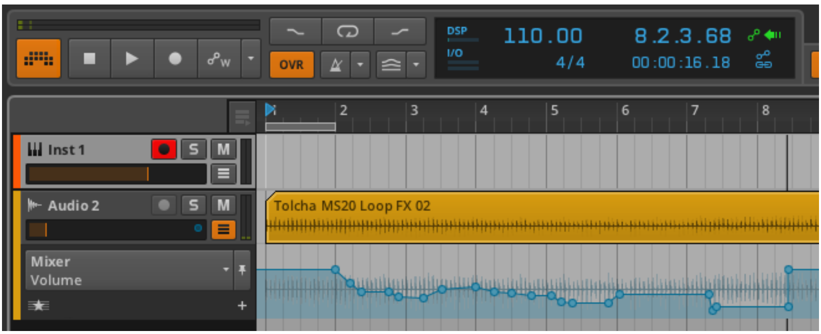
› *Write* mode is the most destructive, recording automation values from the time the transport is launched until it is stopped. Any preexisting automation points that are passed will be overwritten.

Automation recording is separately included in both the **Arranger Timeline Panel** and the **Clip Launcher Panel**.

To record automation in the Arranger Timeline: enable the Automation Record button in the transport controls section of the window header, and then start the transport.



Whether the transport is playing or recording, any parameter adjustments made on this track will be printed as automation. Once the transport is stopped, the automation curve will be optimized and the Automation Record button will be disabled.



Note

If the *Options › Write Automation on Record* option is enabled, the Arranger's Automation Record button will automatically be enabled whenever the Global Record button is armed.

To record automation in the Clip Launcher Panel: enable the track's record arm button and the Launcher's Automation Record button, and then begin recording a clip.

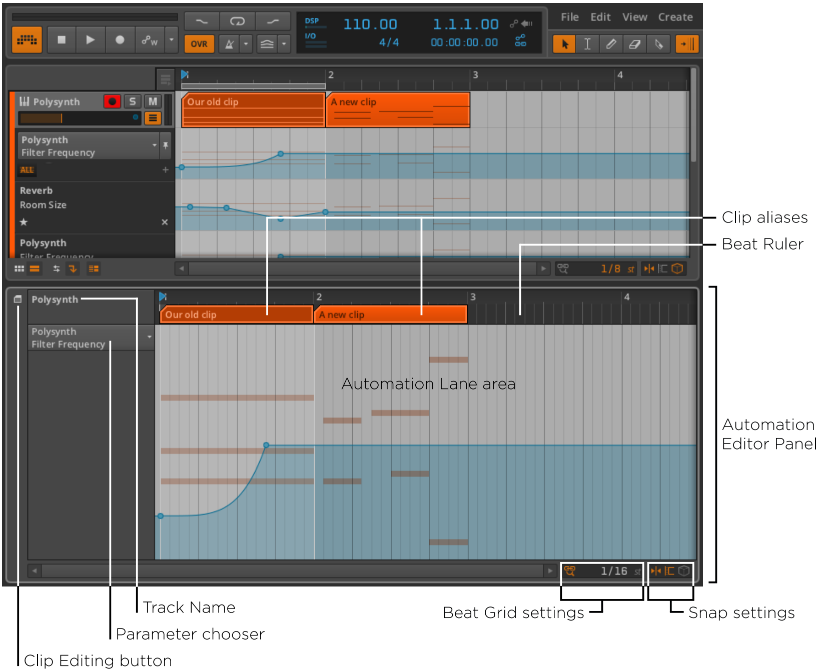
If the Automation Lane button is enabled for the track, the clip's automation will be displayed in the bottom of the clip.

8.2. The Automation Editor Panel

Each panel in Bitwig Studio is focused as narrowly as possible on a specific function. The **Arranger Timeline Panel** is, by necessity, the broadest of our editors. While it also supports working with automation, that is not its primary purpose. Working with automation is, however, the only purpose of the **Automation Editor Panel**.

8.2.1. Track Editing Mode

When the **Automation Editor Panel** is initially called up within the **Arrange View** (by clicking the **Automation Editor Panel** button in the window footer), it opens in *track editing mode*.



In this mode, the interface should look quite familiar. Due to the presence of the *Beat Ruler* (see section 3.1.1), unique *beat grid settings* (see section 3.1.2), and unique *snapping settings* (see section 4.2.2), this looks a lot like the **Arranger Timeline Panel**. The difference is that the general purpose Arranger Timeline area has been replaced with the *Automation Lane area* for our currently selected track.

And the Automation Lane area is essentially an enlarged version of the primary automation lane we just saw in the **Arranger Timeline Panel**. This one also has a Parameter chooser on the left side, and the Automation Lane area is being used to display the automation curve of this parameter over a backdrop of the track's contents.

All of the automation drawing and editing functions we learned in the Automation Lane section of the **Arranger Timeline Panel** will work identically here. But there are a couple differences.

› The **Automation Editor Panel** contains only one automation lane. If you are looking to view multiple parameters from one track, the **Arranger Timeline Panel** is the way to go.

› The *clip aliases* (that float above the Automation Lane area in the Beat Ruler) are indicators of where the track's clips are starting and ending. But these aliases are also editable.

  In the same way that Arranger clips can be moved (see section 4.2.2), edited (see section 4.2.3), and looped (see section 4.2.4), these same actions will work on the clip aliases. Just remember that the Automation Follow setting (see section 8.1.3) will determine how automation is affected by any clip movements or edits.

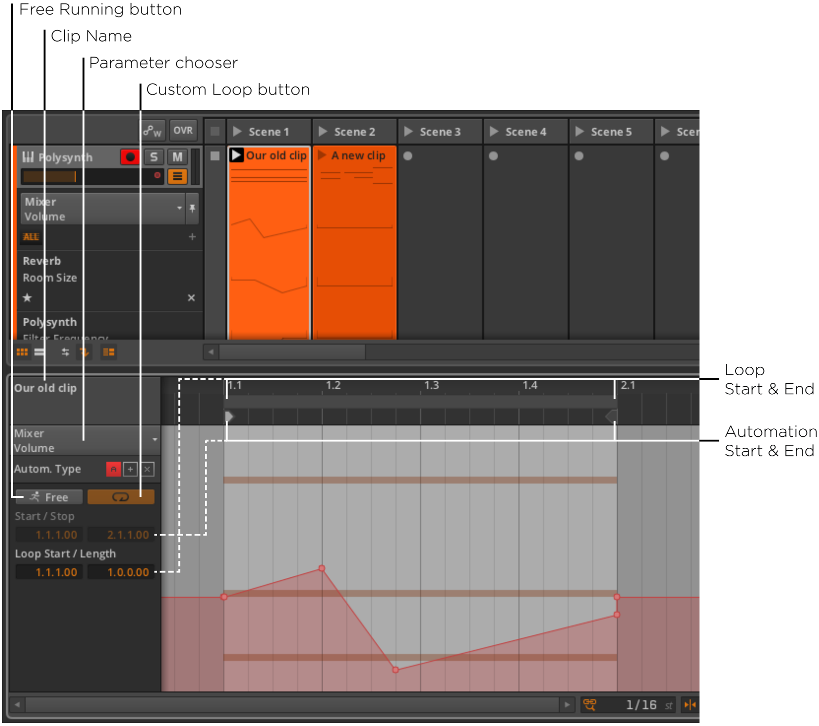
So this track editing mode of the **Automation Editor Panel** is a focused way to work with standard track-based automation. And for less standard, less track-based automation, there is the *Clip Editing button* in the top left of the panel.

8.2.2. Clip Editing Mode

At times it will be useful to have automation attached to a clip rather than to a track's timeline. This is ideal, for example, whenever you want the automation to repeat each time the clip does, or when you are working with the Clip Launcher.

When you want automation to be attached to an Arranger clip instead of the track's timeline, you can switch the **Automation Editor Panel** from track editing mode to *clip editing mode* by enabling the *Clip Editing button*.

When you are working with Launcher clips, all automation is done in clip editing mode with the **Automation Editor Panel**.



Once we break out of the track-based mindset, the same considerations come up as when we talked about the **Clip Launcher Panel** originally. Without the context of a track, our clips are essentially untethered to any fixed timebase or duration. And for this reason, clips viewed here generally use position *1.1.1.00* (often spoken as "bar 1, beat 1") as the relative start of the clip.

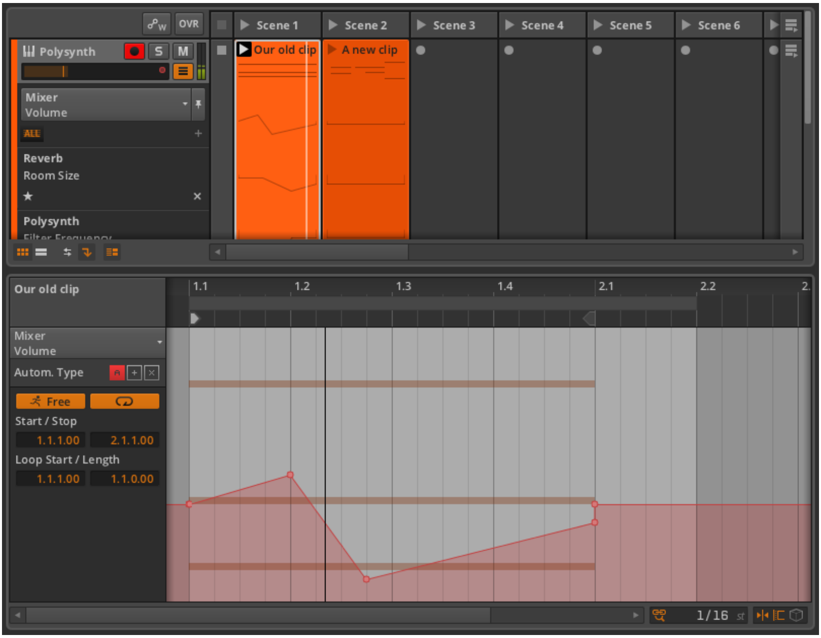
This is also where the Launcher's notion that clips should loop by default comes into play. In the *clip editing mode* of the **Automation Editor Panel**, we now get to decide if a clip's automation data should be tethered to its musical content or should play more freely.

The *Free Running button* contains an icon of a man running with the word *Free*. Once enabled, the clip's automation data can now be adjusted to play back differently from the clip's notes/audio. Once the Free Running button is enabled, the *Start* parameter below can now be adjusted, determining which part of the clip's automation will play back first.

Beside the Free Running button is the *Custom Loop button*. When enabled, this allows you to set different values for the automation's *Loop*

Start and loop *Length* settings. When disabled, the clip's automation will loop just as the clip's musical content does.

These options can create some very dynamic situations, as in the example shown below.



Aside from the Free Running and Custom Loop buttons being enabled, the only change made was increasing the automation's loop *Length* from *1.0.0.00* (one bar) to *1.1.0.00* (one bar and one quarter). By making the automation loop repeat every five beats while the clip's notes repeat every four beats, the automation and notes will only line up in every fifth bar (every 20 beats).

Note

When any of these parameters are changed, you will need to retrigger the clip for the changes to be registered.

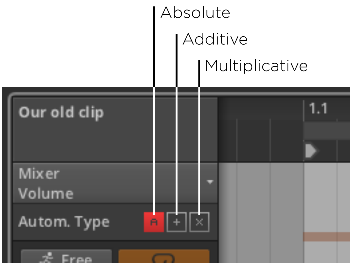
This example is just one way to create rich variation among a single clip's musical content and automation. With the options available, you are free to find your own preferred usage.

8.2.3. Relative Automation

All the work we have done so far involved *absolute automation*. In this paradigm, the automation values specified map to exact values in the parameter's units. A series of examples was already given at the top of this chapter: *-9.43 dB*, *2.88 kHz*, and *124 %*.

Bitwig Studio also has the capability to adjust most parameters in a relative way. With *relative automation*, you can move a parameter ±50% of its total range (*additive automation*), or scale a parameter toward zero, anywhere from 100% of its current value to 0% (*multiplicative automation*).

When we started working in clip editing mode, three buttons appeared beside the *Autom. Type* label.



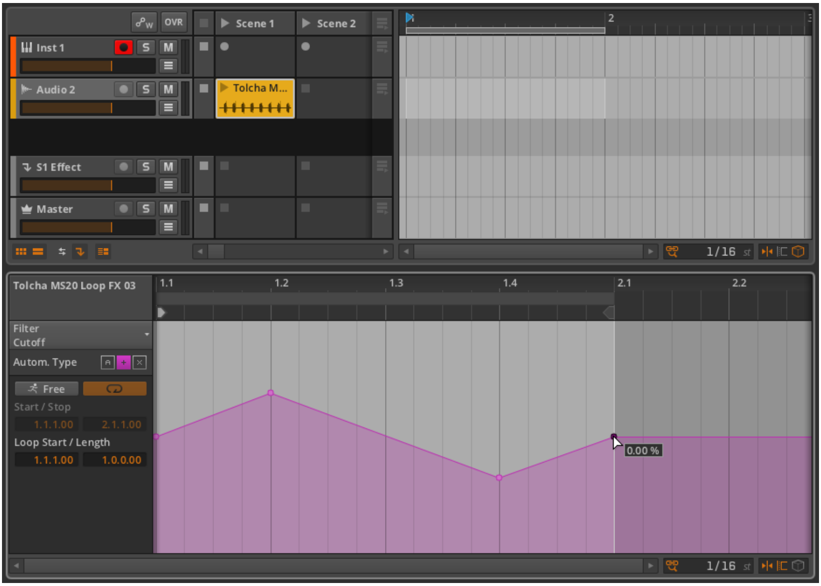
These three icons represent our automation mode choices of *absolute automation* (*A*), *additive automation* (*+*), and *multiplicative automation* (*x*).

When any of these icons are shaded in, this indicates the presence of that type of automation. So the image above is displaying that absolute automation was present for the selected parameter. An unshaded icon suggests that none of that automation type is present.

Note

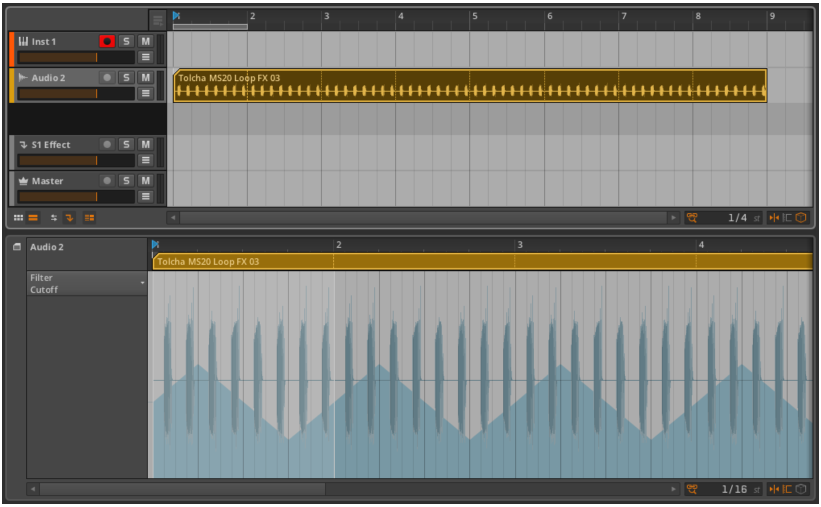
All forms of automation can be present for a single parameter. In this case, the absolute automation is applied first and then modulated by the additive automation. Multiplicative automation is applied last and has the final word, as multiplication always does.

For one example use, I will take a one-bar Launcher clip. I want its filter cutoff to move up a little, down a little, and then back to the middle in each bar. I can draw this with additive automation.

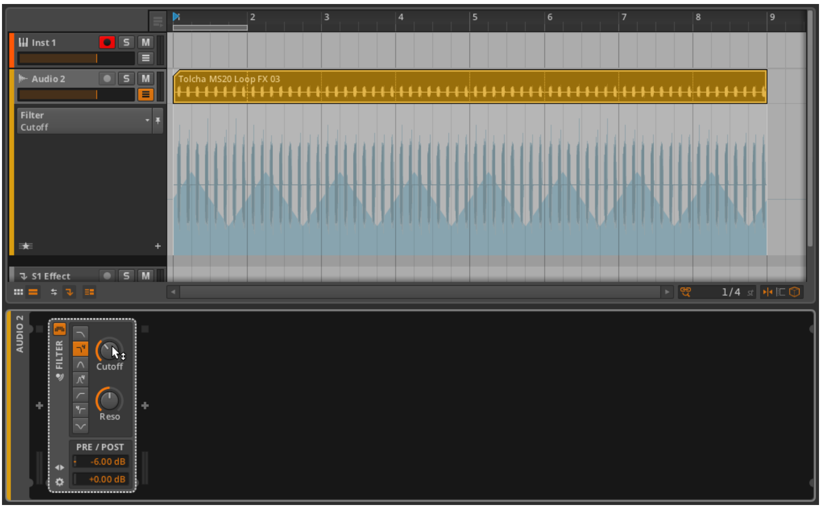


We can see that this automation is ending at *0.00%* so this additive automation is bipolar, moving up to about *20.0%* and dipping evenly to about *-20.0%*. We can also see that the additive modulation icon is the only one shaded so it is currently the only form of automation for this parameter.

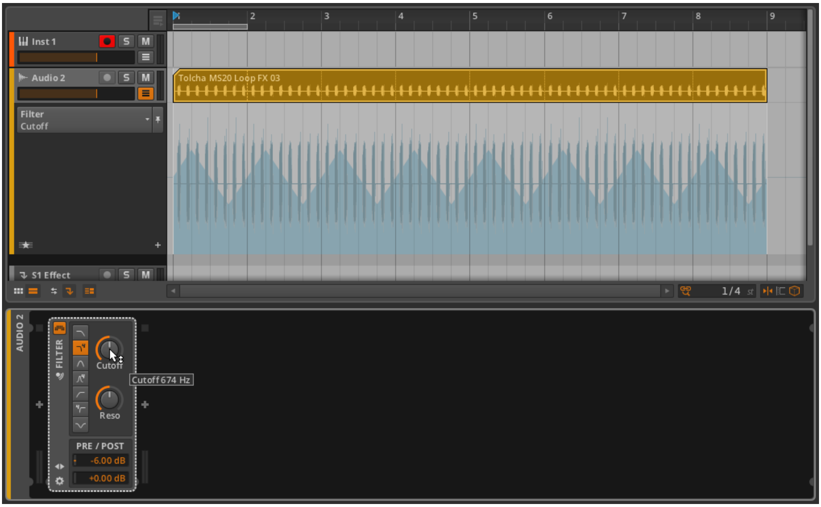
Next, I will drag this Launcher clip into the Arranger and loop it so that it lasts for eight bars.



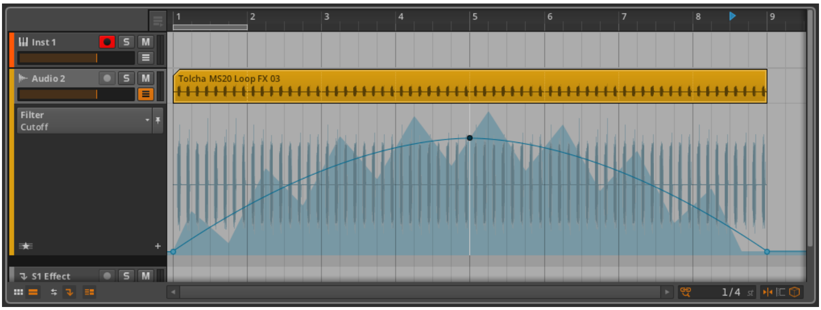
By viewing the absolute automation — the track automation here, since we are back in the Arranger — the automation curve has been extend for our eight bars, but it doesn't appear to be balanced around zero anymore. Let's look at both our automation and **Filter** device together.



We can see now that the default value of the *Cutoff* parameter is a good deal below the center of the range. Since the automation is relative, we can move the *Cutoff* knob to recenter where the automation lands.



I will leave you here with two ideas. The first idea is to now draw absolute automation over the course of these eight bars, taking the *Cutoff* from low to high and then back to low. I will do this by double-clicking to add three automation points, and then [ALT]-clicking and dragging the center point to reshape the curve.



The solid blue line represents the absolute automation curve. The shaded curve is showing the final parameter value, which is the result of both the absolute and relative automation together. By activating the transport, you would see the *Cutoff* control animated to match the absolute automation curve, and the *Cutoff* knob's indicator ring would be moving to match the final parameter value.

The second idea is to not use absolute automation. Instead, use relative automation to give a sense of motion. And then during playback move the parameter control itself in realtime, perhaps with a MIDI controller (see chapter 13). This could be a very strong performance technique.

Note

Whenever a parameter's level indicator is moving separately from its control (as with the *Cutoff* knob and its indicator ring in the previous example), *modulation* is taking place. Relative automation is one form of modulation, and several others are discussed in section 14.2.

9. Working with Audio Events

We spent a healthy amount of time in the early chapters of this document talking about clips and their centrality to music production in Bitwig Studio. Even as the last few chapters have focused on other facilities of Bitwig Studio, clips are still a central part of the conversation. They are the vessels which hold our musical ideas, allowing us to manage, manipulate, copy, and vary these fragments into something greater.

And while we can call the clip our "musical atom," science tells us that atoms are made up of even smaller pieces and particles. In this chapter and the next, we will discuss the audio events and note events that clips are made of. (Whenever we refer to the "musical content" of clips, we are referring to the same audio events and notes.)

We have already examined the various capabilities for manipulating whole clips, whether they are Arranger clips (see section 4.2.5) or Launcher clips (see section 5.2.3). By using the **Detail Editor Panel**, we will begin working at the event level and seeing what tools are available to us at this deepest level of musical arrangement. And once we couple that interface with the **Inspector Panel**, most of the editing options and optimized workflows offered by Bitwig Studio will now be at our fingertips.

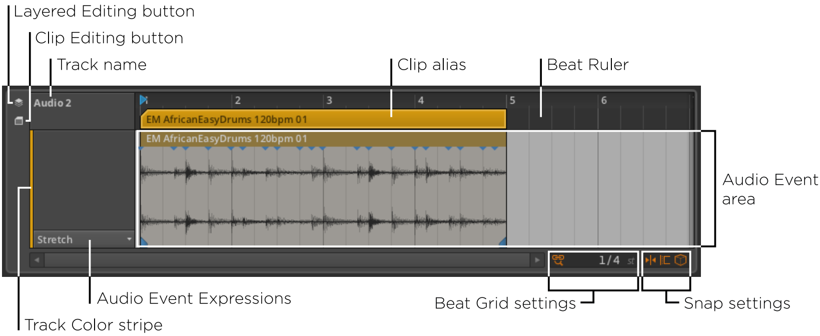
So let us begin the detail work of creating and preparing music. Next stop: *audio events*.

9.1. The Detail Editor Panel, Audio Clip Edition

All music is assembled in clips in Bitwig Studio. Just as a primary purpose of the **Automation Editor Panel** is to work with various kinds of clip automation, the purpose of the **Detail Editor Panel** is to work with the musical content of clips.

9.1.1. Layout of the Detail Editor Panel

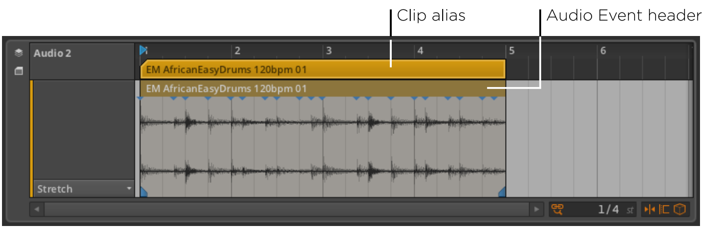
By double-clicking a clip, the **Detail Editor Panel** will be called up with its focus on that clip. For the examples in this chapter, we will use audio clips, and we will start by double-clicking an audio clip from the Arranger Timeline.



After working with the **Arranger Timeline Panel** and the **Automation Editor Panel**, many of these interface elements should be familiar, including the *Beat Ruler* (see section 3.1.1), the *clip aliases* (see section 8.2.1), this panel's own *beat grid settings* (see section 3.1.2), and the *snapping settings* (see section 4.2.2). Even the currently inactive *Clip Editing button* (see section 8.2.2) is here, indicating that we are starting in *track editing mode*.

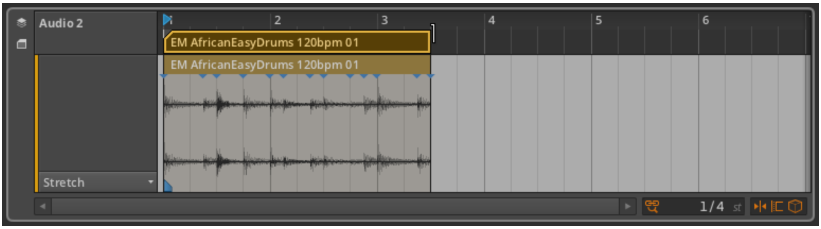
But as with the previous timeline-based panels, the sections that have changed are substantial and unique to the operation of this panel.

The central *Audio Event area* is where all audio events are displayed in this panel. Audio events that appear here have their own headers, which can look redundant right below the clip's alias.



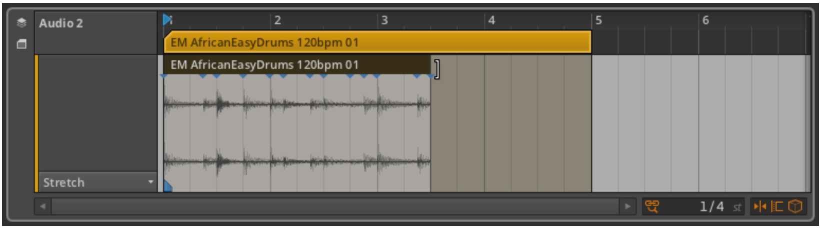
One example will illustrate the relationship between the clip and the contained audio event.

To adjust the length of a clip: mouse over the top right edge of the clip alias so that a half-bracket cursor appears. Then click and drag the mouse horizontally.



By shortening the clip, you can see that the audio event is also shortened. The clip is the parent in this relationship, and the children (audio events, in this case) can exist only where the parent is there to allow it.

To adjust the length of an event: mouse over the top right edge of the event so that a bracket cursor appears. Then click and drag the mouse horizontally.

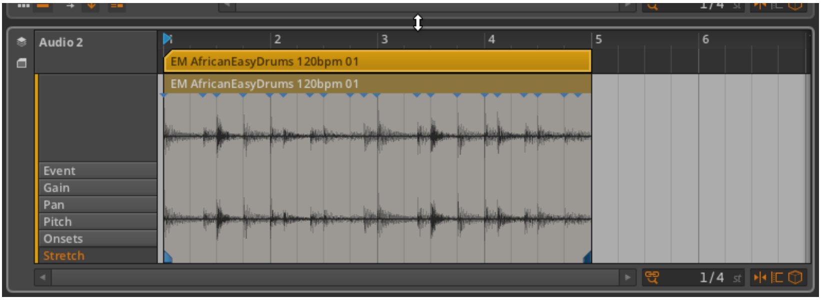


By shortening the event, you can see that the clip itself is unaffected. You can do anything you want with this empty clip space: insert a short audio event/sample, duplicate as much of the previous event as will fit, or leave it blank as a rest. Nothing placed in the clip will be allowed to go beyond its boundaries, but all the available space can be used.

As you may also have noticed, no looping cursor appears when navigating the audio event's header. Clips are the smallest units where most arranging tasks are carried out. Accordingly, looping can be applied as an arrangement gesture for clips, but not for audio events (or notes).

9.1.2. Audio Event Expressions

To the left of the Audio Event area is a space for specifying which *audio event expression* is being displayed — and potentially edited. The images shown a moment ago displayed a menu in this area. But if you prefer a list of all available audio event expressions, drag the top border of the **Detail Editor Panel** so that it grows.

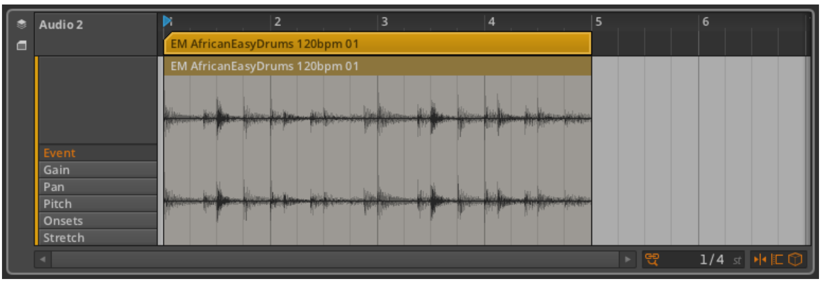


Audio event expressions — also called *expressions* — are parameters that can be set within each individual audio event. Several of these parameters can change over the course of the event, making them like specialized automation curves. Others are a series of location markers that are used to affect the playback of the audio event.

Only one expression can be focused on at a time, and you pick which expression to view by clicking its name in the list. We will examine them in order, starting at the top of the list.

9.1.2.1. Event Expressions

The *Event* presents a simple display of all audio events.

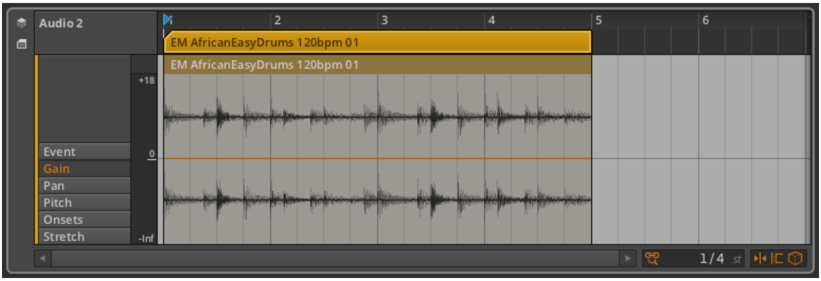


No actual expression curve or other data is shown here. This allows you to freely move and edit the audio events themselves without inadvertently changing other values.

Audio events are moved and adjusted in the same way as regions (see section 4.2.3) except that the range of motion is limited to the length of the parent clip. When compared to the **Arranger Timeline Panel**, all tools function equivalently in this panel except for the pencil tool.

9.1.2.2. Gain Expressions

Gain expressions represent a level control for the audio event.

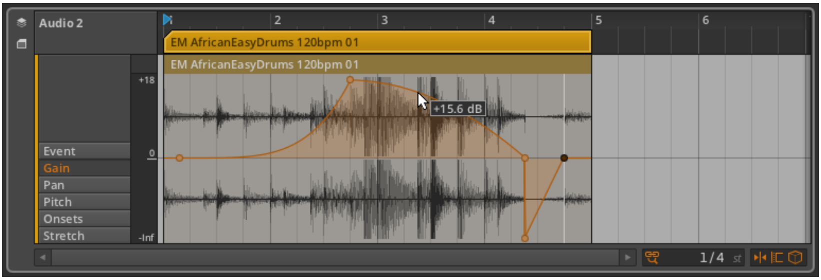


This expression can be made up of a series of points that are created and edited in the same way that automation points are (see section 8.1.2).

The gain expression is measured in units of decibels with the center line representing zero decibels of change (unity gain).

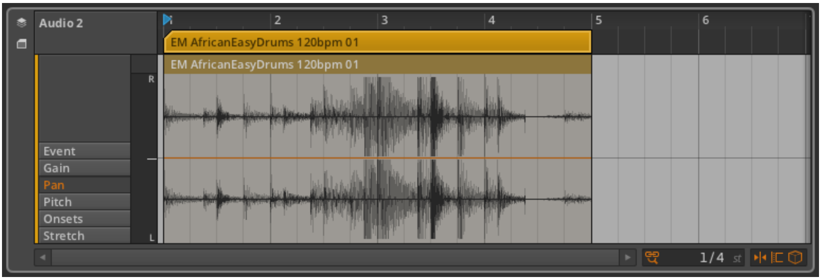
A gain expression is identical in function to volume automation. The difference is that the expression is applied to the audio source itself, and volume automation is applied as the last stage of a track's signal flow (after the track's device chain and everything else).

Since the gain expression affects the source material, the waveform is helpfully redrawn to show the effect of this expression.



9.1.2.3. Pan Expressions

Pan expressions represent a stereo placement control for the audio event.



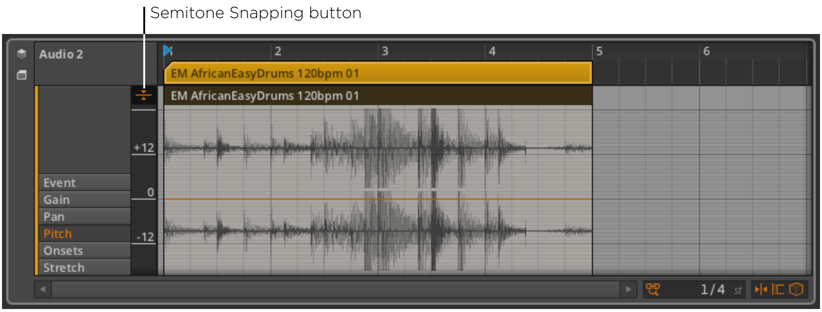
This expression can be made up of a series of points that are created and edited in the same way that automation points are (see section 8.1.2).

A pan expression is measured as a bipolar percentage with the center line at *0.00%* (center placement, or no panning adjustment), *100%* for hard right, and *-100%* for hard left.

As with the gain expression, the *pan expression* is applied to the audio source itself. This has no direct interaction with *pan automation*, which is applied by the track mixer after the device chain.

9.1.2.4. Pitch Expressions

Pitch expressions represent a frequency transposition control for the audio event.



Note

This expression will take effect only with certain audio event playback modes (see section 9.2.1.2).

This expression can be made up of a series of points that are created and edited in the same way that automation points are (see section 8.1.2).

A pitch expression is measured in semitones (or half steps) with the center line at *0.00* (zero semitone shift for no transposition), a maximum of *24.00* (two octaves up), and a minimum of *-24.00* (two octaves down).

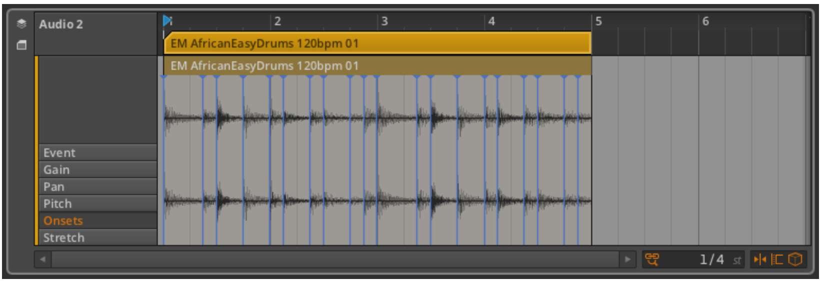
Note

Unlike the other expressions, the pitch expression's vertical axis is scrollable and zoomable (by clicking and dragging it). Because of this, it will not automatically compact itself to fit a small **Detail Editor Panel**.

The *semitone snapping* option causes pitch point changes to snap to whole number semitones. As with the position snapping options (see section 4.2.2), holding [SHIFT] will toggle this behavior.

9.1.2.5. Onsets Expression

The *Onsets* expression represents locations in an audio event where the sound's envelope substantially changes, often where individual sounds occur.



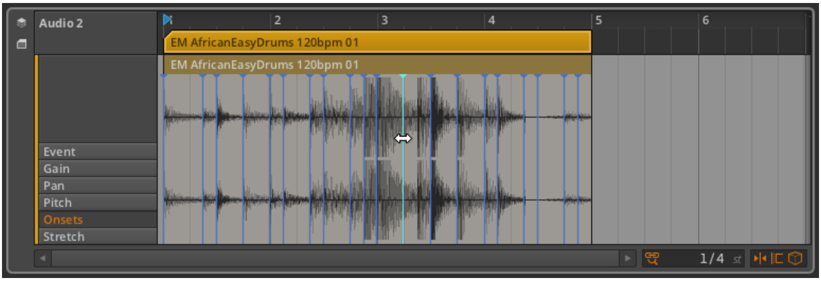
Onsets are used both as data to help preserve the sound quality of single audio events, and as demarcations when splitting the component parts of one event into multiple, individual events.

When a sample is initially dragged into a Bitwig Studio project, it is analyzed for its tempo, its musical length, and where onsets occur in the file. Each onset is represented by a vertical blue line that reaches a small blue triangle at the top of the event.

You can also manually insert or manipulate onsets, either because the automatic results were imprecise or to manipulate how stretching is done during playback (see section 9.1.2.6), etc.

To insert an onset: double-click any area of the event away from a current onset.

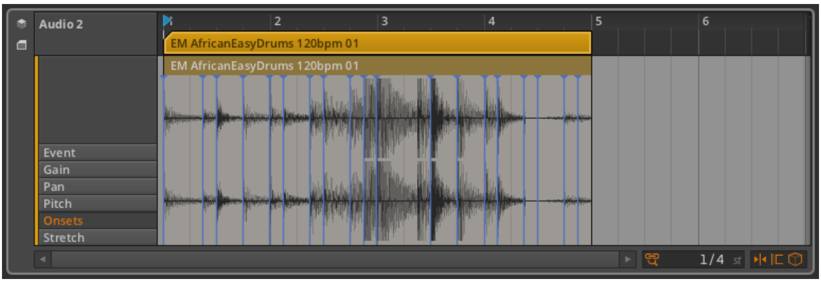
To move an onset: click and drag the point with the mouse.



Note

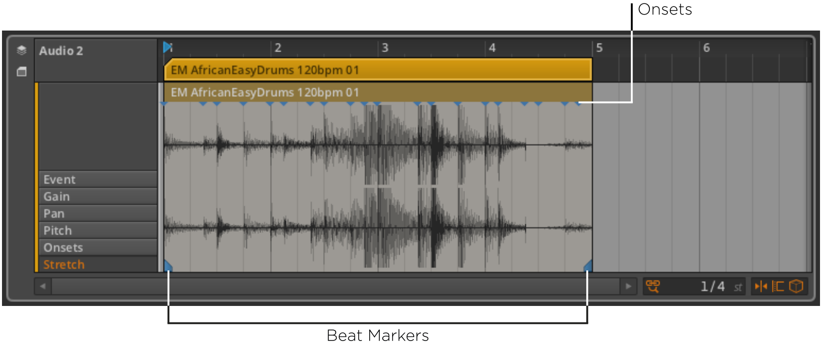
Onsets are usually colored a periwinkle shade of blue. Selected onsets are tinted a bright cyan.

To delete an onset: double-click it. Or single-click the point to select it, and then press [DELETE] or [BACKSPACE].



9.1.2.6. Stretch Expressions

Stretch expressions determine how the playback speed is altered, thereby stretching the audio file.



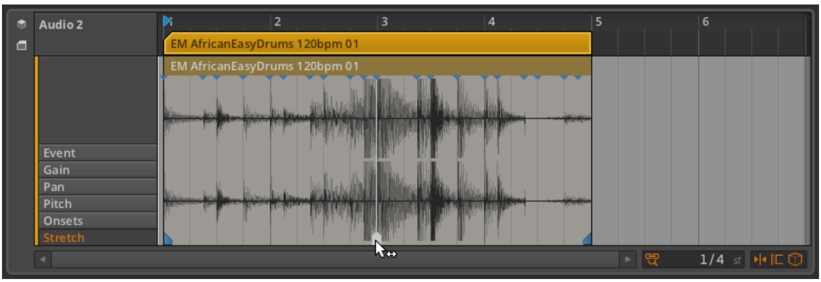
Note

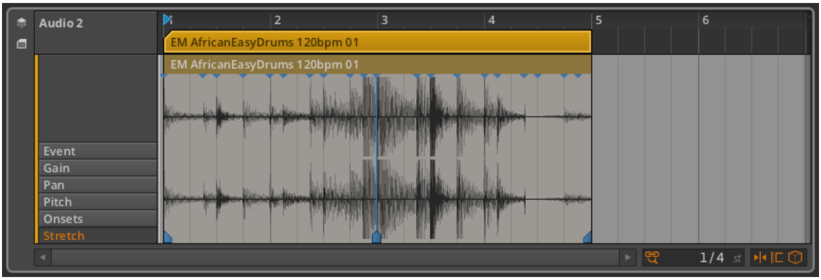
This expression will take effect only with certain audio event playback modes (see section 9.2.1.2).

The stretch function of this expression is achieved by inserting *beat markers*, which dictate the points in the audio event that are locked to their position. The playback speed of the area between beat markers is then altered to ensure that those beat markers occur at their assigned times.

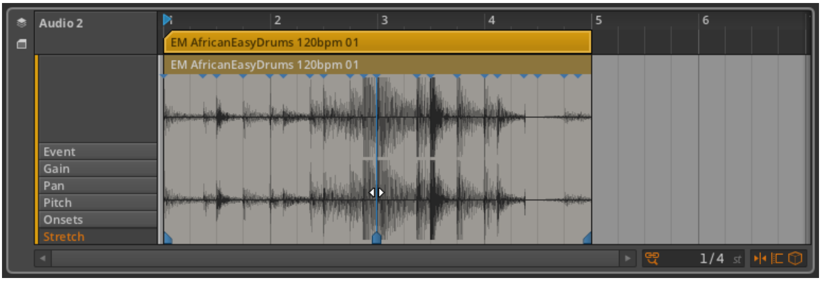
By default, only the start and end times of each event are given beat markers, but the stretch expression makes it easy to create a beat marker where an onset already exists.

To create a beat marker: double-click any area of the event. Or mouse around the bottom of the event, and then single-click any white marker that appears.

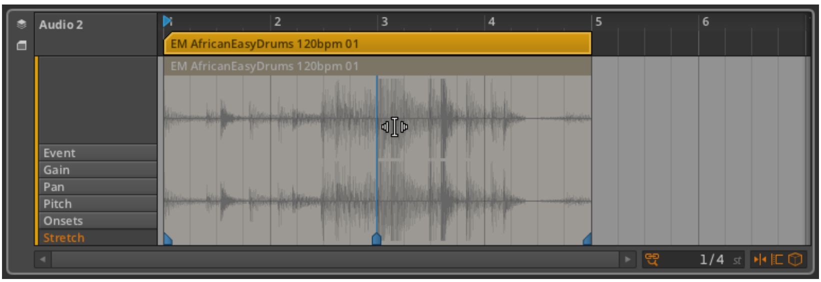




To move a beat marker and its surrounding audio: along the bottom half of the event, click and drag a beat marker with the simple, double-arrow cursor.

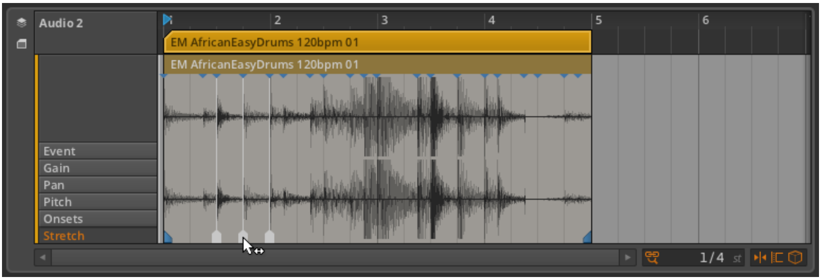


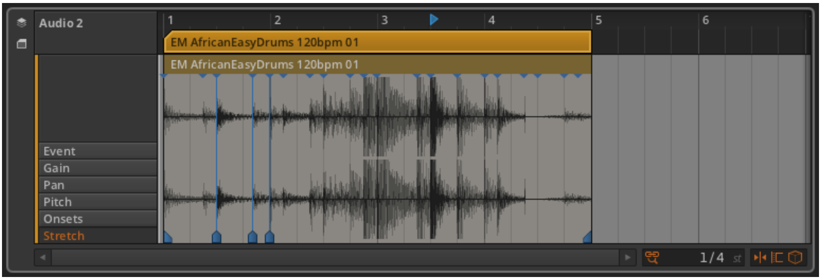
To keep a beat marker in place and fine-tune the position of the audio around it: along the top half of the event, click and drag a beat marker with the radiating I-beam cursor.



The combination of moving beat markers and then "sliding" them precisely will speed up any workflow involving audio stretching.

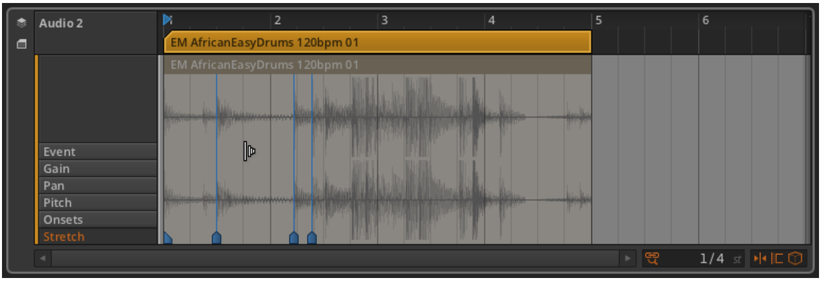
To convert a trio of onsets to beat markers: hold [ALT] and mouse around the bottom of the event until the desired three white markers appear. Then click and drag the mouse horizontally.





This allows you to stretch a particular area of your audio event while keeping the rest of the event unaffected.

To freely stretch the size of a region: [CTRL]+[ALT]-click ([CMD]+[ALT]-click on Mac) a region and drag horizontally.



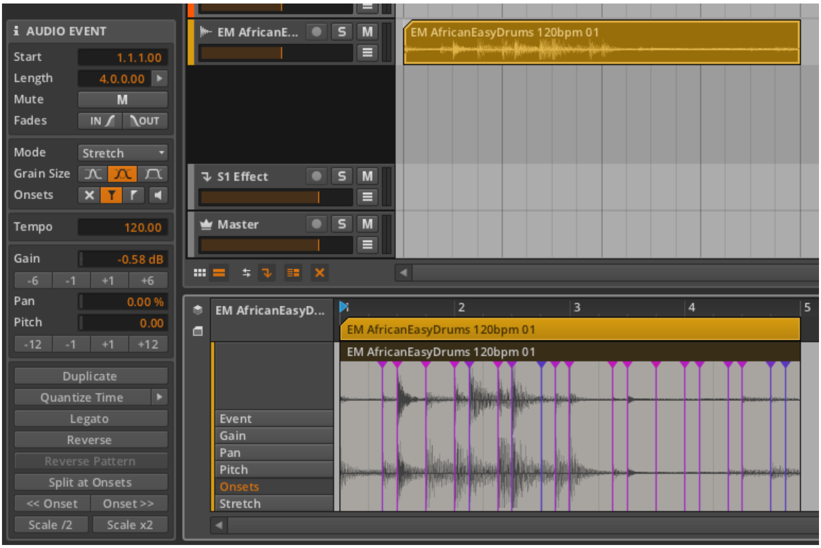
Be careful as this will move beat markers around. You can also access this function by clicking and dragging the start or end of any audio event.

9.2. Inspecting Audio Clips

As was said in this chapter's introduction, we have been using the **Inspector Panel** to examine clips for quite some time. In addition to the clip settings we have already examined, any non-empty clip has a large section at the bottom of the **Inspector Panel** for dealing with its musical contents.

9.2.1. The Inspector Panel on Audio Events

By selecting a clip, certain parameters are revealed in the *AUDIO EVENT* section, but when selecting an audio event itself (by single-clicking the audio event's header in the **Detail Editor Panel**), the **Inspector Panel** provides all settings and functions relevant to the selected event(s).

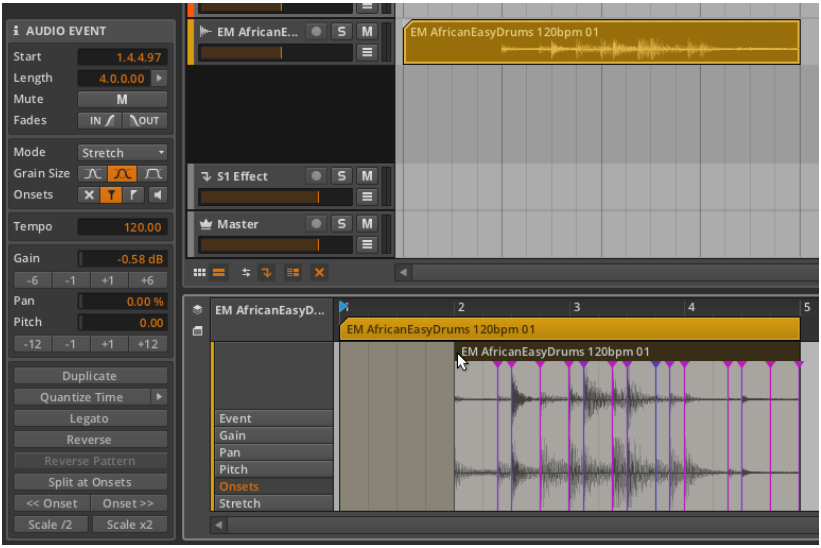


Several of these settings will be familiar. Since there are many of them, we will take them one section at a time.

9.2.1.1. Timing and Fades Section

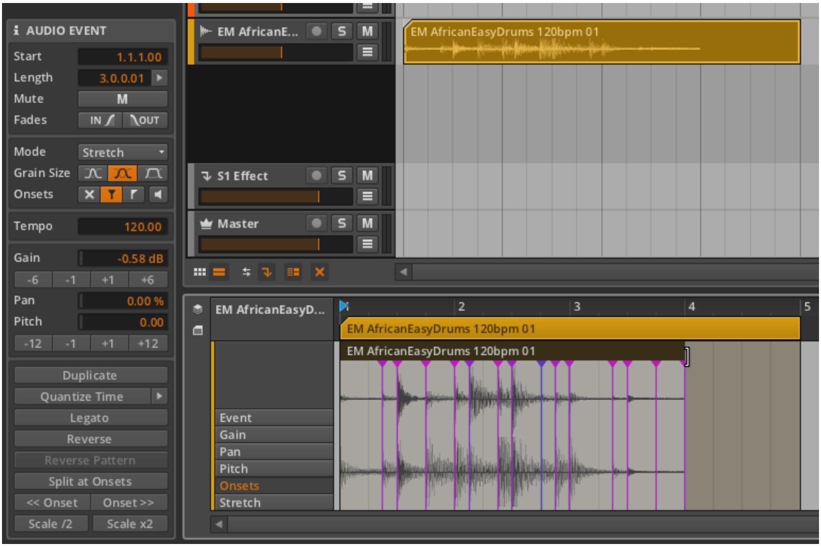
These settings generally relate to the musical position of the selected event and its optional fades:

› *Start* sets the start position of the event within its parent clip or track. Adjusting this position will move the audio event as it exists, the same as clicking and dragging the event within the **Detail Editor Panel**.



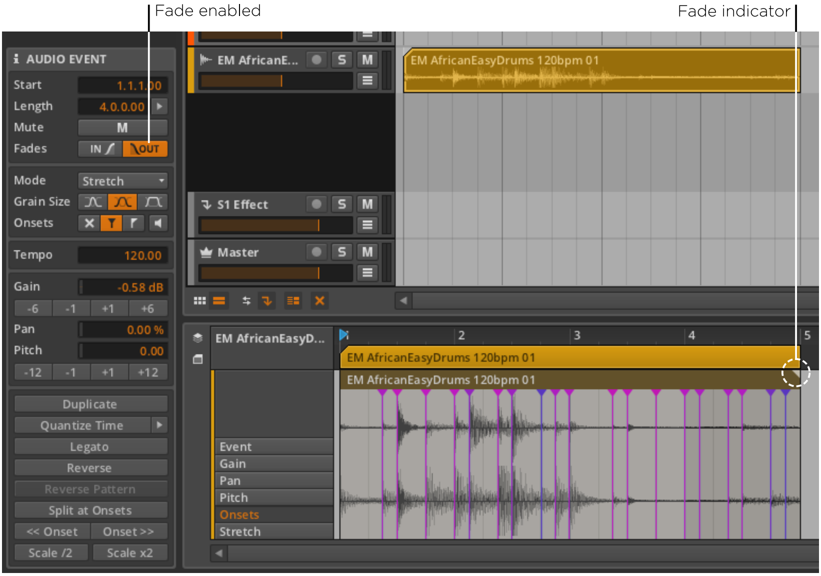
> **Note**
>
> Remember that audio events will be always be truncated by the boundaries of their parent clip.

› *Length* sets the duration of the event within its parent clip. Adjusting this duration will simply lengthen or shorten the event, the same as using the bracket cursor to adjust the right edge of the event's header.



› *Mute* toggles whether or not the event is disabled on playback.

› The *Fades* buttons toggle a fade *IN* and fade *OUT* for the event. The purpose of these quick fades is to avoid clicks that can happen when an audio file is split.



The event header is given a triangle in its top left or top right corner to indicate a fade in or a fade out, respectively.

9.2.1.2. Stretch Section

These settings relate to the behavior of Bitwig Studio's audio playback:

› *Mode* sets the audio playback algorithm for the audio event. The settings include:

  › *Raw* ignores all stretch expression data. Events are played back at their original speed, regardless of the project tempo or any other considerations.

  - *Stretch* is an optimized algorithm that time-stretches audio to match the project's tempo. This algorithm also allows pitch and playback speed to be set independently.

  - *Stretch HD* is a similar algorithm to *Stretch*. *Stretch HD* is tuned with higher quality in mind, which can put more load on your computer's processors.

  - *Repitch* ties pitch and playback speed together (as a tape recorder would). Stretch expression data is respected while pitch expressions are ignored.

- *Grain Size* adjusts the length of each audio segment that is stretched in the selected audio event. The three relative options are for either short, medium, or long portions of the audio to get processed at a time.

- *Onsets*controls how the onsets expression (see section 9.1.2.5) is used to adjust playback. There are three options to choose between and one optional mode:

  - The first option is *off*, represented by an *x* icon. In this mode, the onsets expression is completely ignored for playback purposes.

  - The second option is *soft*, represented by a centered vertical line with both a "fade out" triangle on the top left and a "fade in" triangle on the top right. This mode emphasizes smoothness by blending the audio before an onset with that that comes after.

  - The third option is *hard*, represented by a centered vertical line with only a "fade in" triangle on the top right. This mode emphasizes rhythmic accuracy by focusing on the audio that comes after the onset.

  - The separate button with the speaker iconrepresents preview mode. When toggled on, this mode plays the audio at each onset, but turns the volume down for all other parts of the event. This is a useful audible indicator of where the onsets are currently placed.

9.2.1.3. Tempo Section

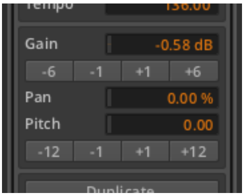
Tempo defines the original tempo of the audio event. Knowing this enables Bitwig Studio to properly play back the data in any circumstance.

When an audio file is brought into a project, the program first checks the filename for an indication of tempo (such as the word *154bpm*). If nothing is found there, the program determines the tempo as best it can.

This value can be corrected at any time, but changing it will impact the placement and timing of the audio event.

9.2.1.4. Expressions Section

This section exposes three of the expressions we have covered: *Gain* (see section 9.1.2.2), *Pan* (see section 9.1.2.3), and *Pitch* (see section 9.1.2.4). While these expressions have completely different functions, they are programmed in the same fashion.



Following the *Gain* and *Pitch* numeric controls are incrementer and decrementer buttons that will adjust the expression value by the declared amount. For the *Gain* expression, these buttons express decibel changes. For the Pitch expression, the unit is semitones.

These are the automation-type expressions, each able to be defined by a curve made from several values. Because of this possibility, each value in this section of the **Inspector Panel** is actually representing the average of points in that expression. Let's examine the *Gain* expression as an example.

The *Gain* value listed of *-0.58 dB* is an average of the five points defined in this audio event expression.

To adjust an expression curve: change its listed average value, or click one of the expression's incrementer/decrementer buttons.

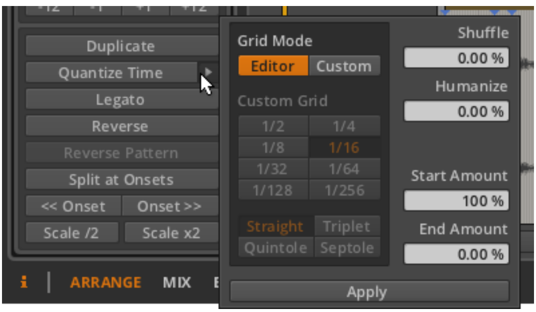
This method will work for any expression in this section, whether it is defined by a curve or a single value.

9.2.1.5. Function Buttons Section

These buttons execute the specified function on the selected audio event(s):

› *Duplicate* places an exact copy of the selected event immediately after it. This function is also available from *Edit › Duplicate Audio Event(s)* or by pressing [CTRL]+[D] ([CMD]+[D] on Mac).

› *Quantize Time* moves the start and/or end times of the selected event in relation to a beat grid. The parameter pane for this function appears when the right-arrow button is clicked.



› *Grid Mode*: Determines whether to adopt the grid settings from the current *Editor* or to allow *Custom* grid settings.

› *Custom Grid*: Exclusive *beat grid resolution* and *beat grid subdivision* settings (see section 3.1.2) for the quantize function.

> **Note**
>
> This is available only when *Grid Mode* is set to *Custom*.

› *Shuffle*: Amount of swing/groove (see section 2.1.1) applied to the beat grid for the quantization function.

› *Humanize*: Amount of randomness added to the quantize function, with the intention of mimicking human imperfection.

› *Start Amount*: Amount of quantization applied to each selected event's start position.

For example, a setting of *50.0%* would move a selected event's start position halfway to the closest grid point. A setting of *100%* places the event exactly on the closest grid point.

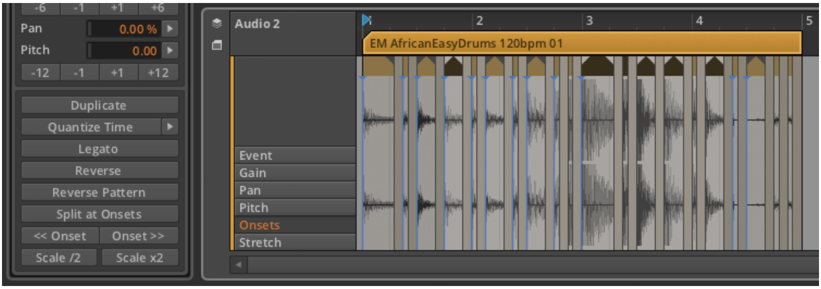
› *End Amount*: Amount of quantization applied to each selected event's end position.

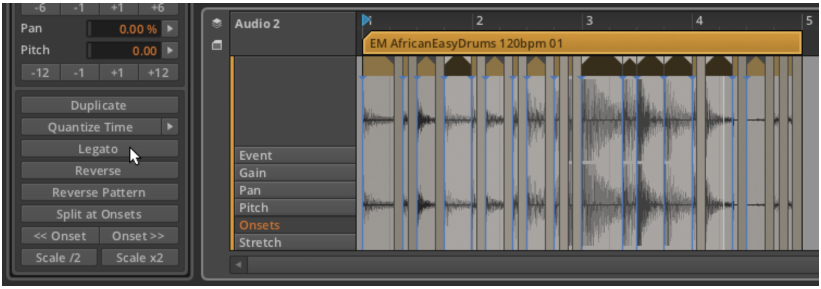
> **Note**
>
> *Humanize* is the last factor applied in the quantize function. So even *Start Amount* of *100%* might not place events directly on the grid if *Humanize* is enabled.

The quantize function can be executed by either clicking the *Apply* button at the bottom of the parameter pane, or by clicking the *Quantize Time* button itself.

› *Legato* adjusts the length of each selected event so that it ends immediately before the next event begins, creating a continuous series of events.

The following images demonstrate a group of selected events both before and after the *Legato* function is applied:





› *Reverse* flips the selected event around, causing it to play backwards. This also flips any event expression curves.

› *Reverse Pattern* flips the order of a group of selected events. This does not cause each event and its expressions to play backwards, but rather causes the last event to be played first, etc.

Note

This function will work only when multiple events are selected.

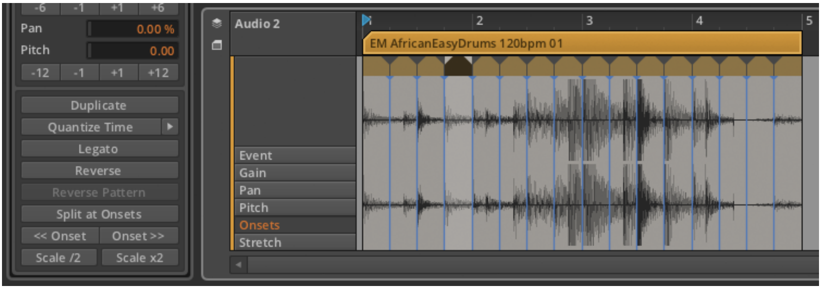
› *Split at Onsets* divides the selected event into multiple events by using each onset as a new event start. This can be an extremely efficient way to do audio edits.

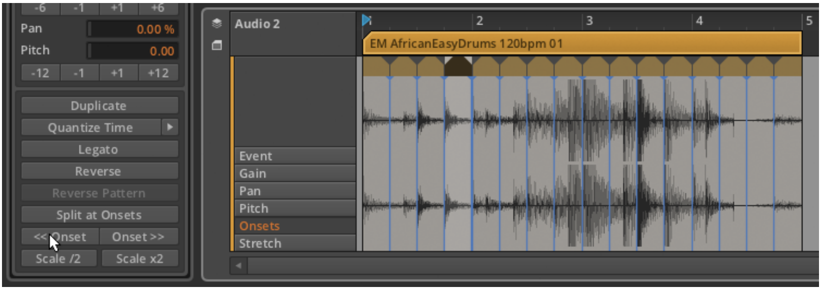
Note

When an audio event is split, either by this function or the Knife tool, fade in and outs will be added to split points if the option *Options › Add Anti-Click Fades on Audio Event Split* is enabled.

› *<< Onset* slides the selected event so that it begins at the previous onset marker, effectively shifting this area to play earlier material. This affects only the content of the selected event.

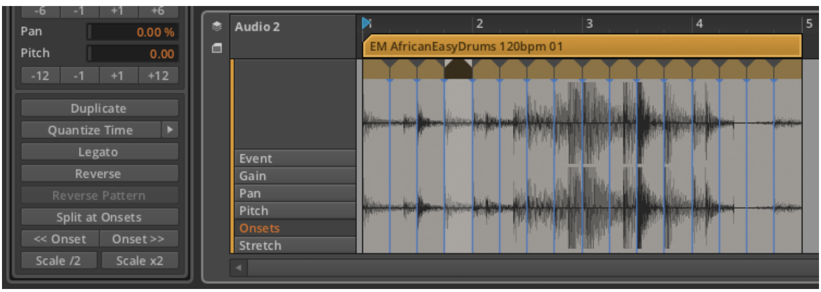
The following images demonstrate a selected event both before and after the *<< Onset* function is applied:

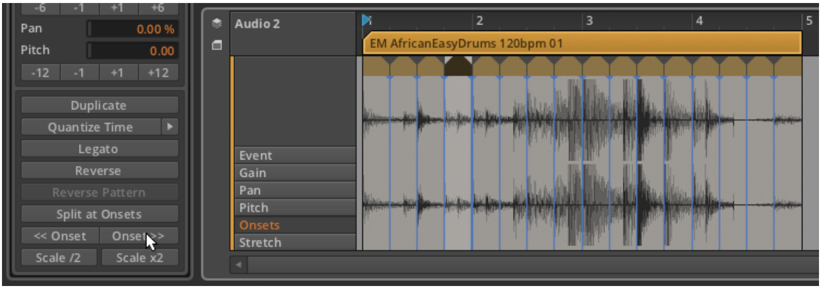




› *Onset >>* slides the selected event so that it begins at the next onset marker, effectively shifting this area to play later material. This affects only the content of the selected event.

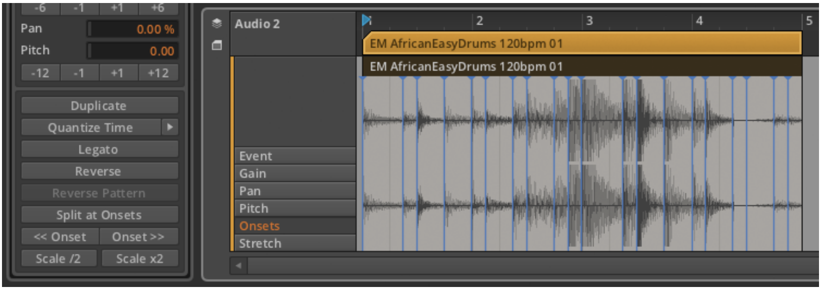
The following images demonstrate a selected event both before and after the *Onset >>* function is applied:

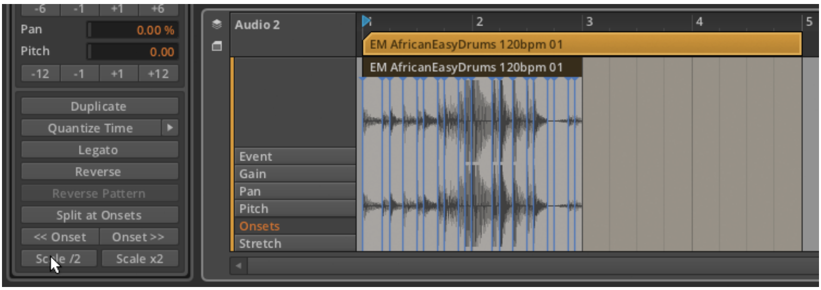




› *Scale /2* halves the length of the selected event, effectively causing it to play back twice as fast. All onset and beat markers are also proportionally shifted.

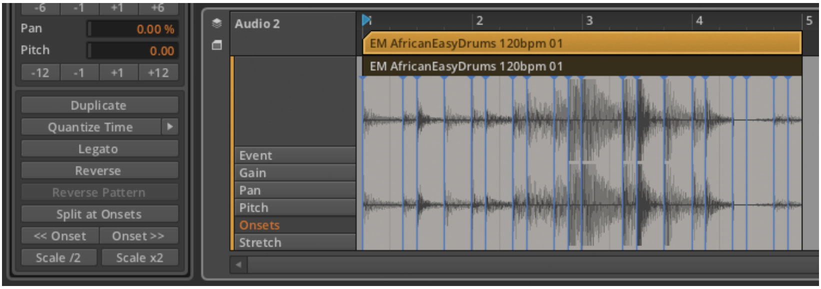
The following images demonstrate a selected event both before and after the *Scale /2* function is applied:





› *Scale *2* doubles the length of the selected event, effectively causing it to play back half as fast. All onset and beat markers are also proportionally shifted.

The following images demonstrate a selected event both before and after the *Scale *2* function is applied:



9.2.2. Working with Multiple Audio Events

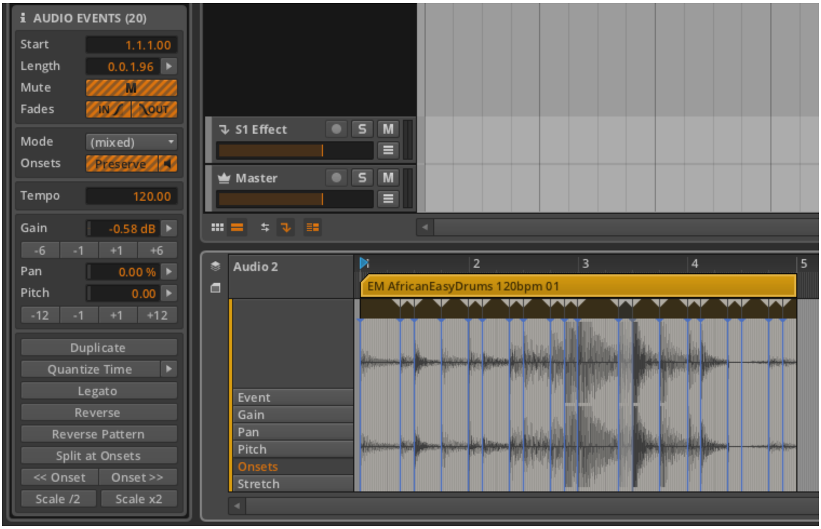
The **Inspector Panel** also works with selections of multiple events.

Functions are straightforward, as most of them listed in this chapter allow the selection of multiple events. (In the case of *Reverse Pattern*, it is not available *unless* you have multiple events selected.)

Parameters can be a little trickier when several events are selected at once. Bitwig Studio has a couple tricks of its own for both displaying and working with chunks of parameter data.

9.2.2.1. Mixed Settings

We saw expressions summarized earlier with a single average of all their points. That works well when you are dealing with numbers, but some parameters simply toggle on and off. For these discrete parameters, the **Inspector Panel** will diagonally stripe any indicator whose settings are mixed.



In the above image, the *Mute*, fade *IN*, fade *OUT*, and both of the *Onset* buttons (*Preserve* and preview) have the orange and gray striping to suggest that some of the selected events are enabled, some are not.

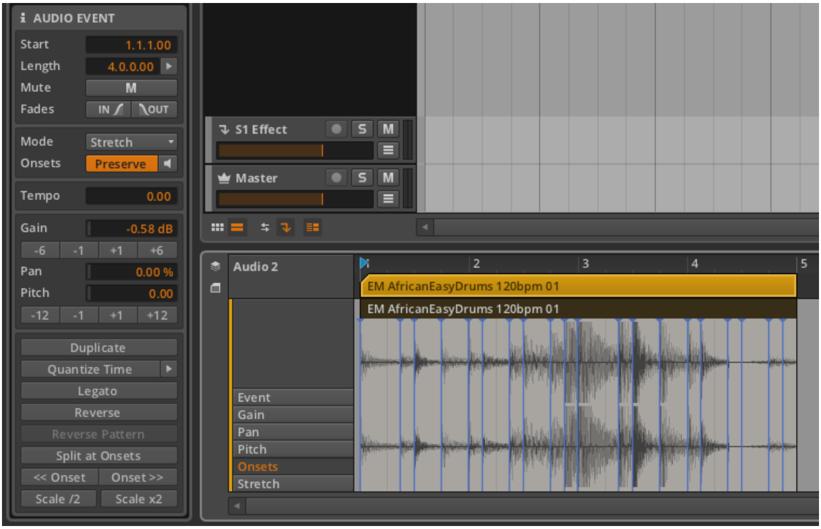
Additionally, the *Mode* menu is listed as *(mixed)*, which is its way of suggesting that not all selected events have a uniform setting.

9.2.2.2. Using the Histogram

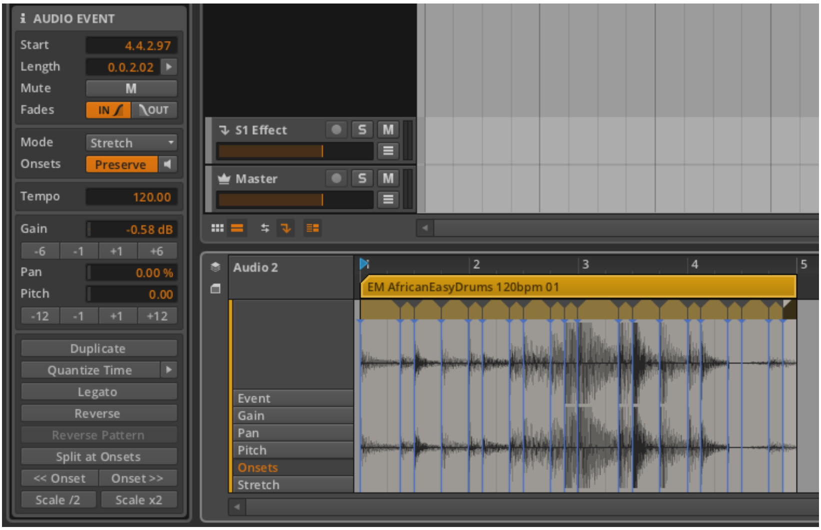
Finally, Bitwig Studio provides a special interface called the **Histogram** for working with a selection of multiple numeric values. The purpose of a histogram is to display the number of times that different possibilities occur over a span of time. In our case, the span of time being considered is the length of the current selection and the possibilities being considered are different values of the targeted parameter.

But our **Histogram** can also modify values, or even produce them from scratch. We will now demonstrate the option of creating values and then tweaking them.

I will begin with the drum loop you have seen throughout this chapter.

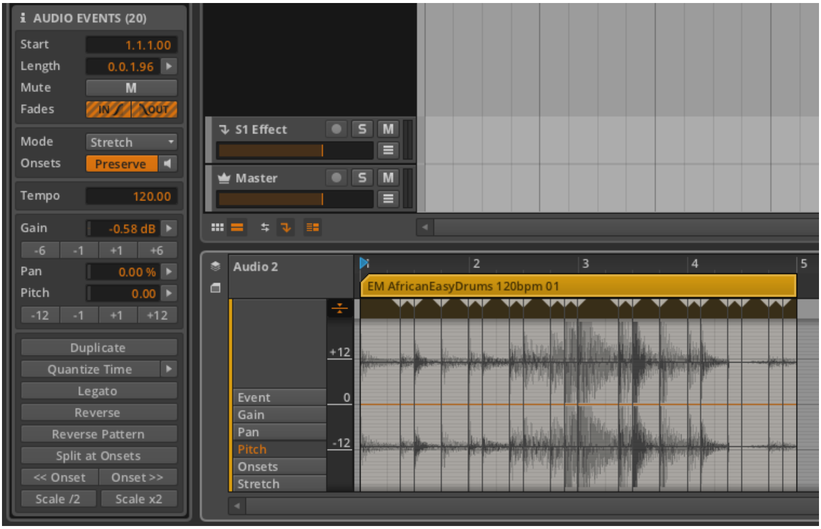


By applying the *Split at Onsets* function, this single event will now be divided at each onset point, giving us a collection of events that add up to the same loop.



From here, I will select all of the events. This can be done in the standard ways, by either pressing [CTRL]+[A] ([CMD]+[A] on Mac), or by choosing *Select All* either from the *Edit* menu or from the context menu.

And once all events are selected, I will switch the **Detail Editor Panel** to focus on the *Pitch* expression.



A few things to note before we proceed.

First, the **Inspector Panel** now labels this section of the panel as *AUDIO EVENTS (20)*. The 20 in the title is indicating exactly how many audio events are currently selected and will be acted upon when changes are made here.

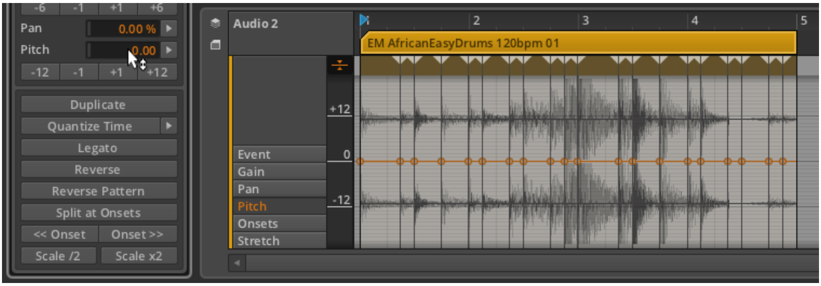
Second, the event headers are now reflecting fades where each onset point was split. This is because I have *Options › Add Anti-Click Fades on Audio Event Split* enabled, which is the default setting.

The only places where fades do not exist are at the start of the first event and at the end of the last one because no splitting occurred at these two places. And because these events lack a fade of each kind, both of the *Fades* buttons are now striped.

Third, in the expression section of the **Inspector Panel**, each numeric control is now followed by a right-arrow button. Since we now have multiple events selected, these arrows appear to give us access to the **Histogram**.

With these few observations made, we can now proceed.

The pitch expression is currently empty, containing no points. Now I will simply single-click on the Pitch parameter control. I am not changing the setting, just clicking on it once.



By just clicking on this parameter, an expression point has been created at the start of each event. So even though every point is currently set to *0.00* (semitones), we now have something to work with.

By clicking the right-arrow button beside the *Pitch* parameter, we can now see the **Histogram**.

The **Histogram** is comprised of four elements:

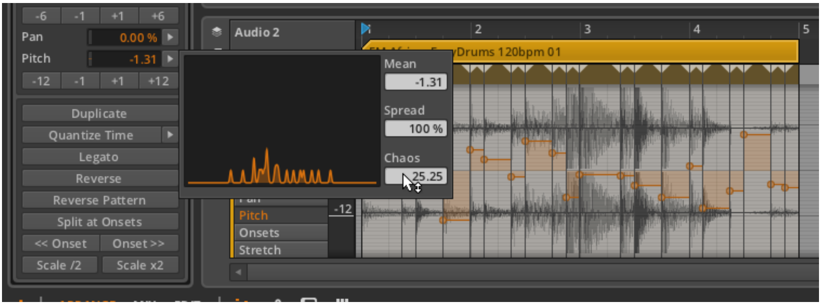
› The large *display* on the left is the actual histogram, which will present a count of the different values occurring across our selection. It is blank right now as we don't have any values yet.

› *Mean* represents the average of all selected values.

› *Spread* is a control for modifying the range of the selected values.

› *Chaos* is a control for injecting random variations to the selected values.

Adjusting the *Spread* of these points would do nothing as they are all currently identical. And adjusting the *Mean* would only adjust them all by an identical amount keeping them the same. So I will click the *Chaos* control and drag it upward.

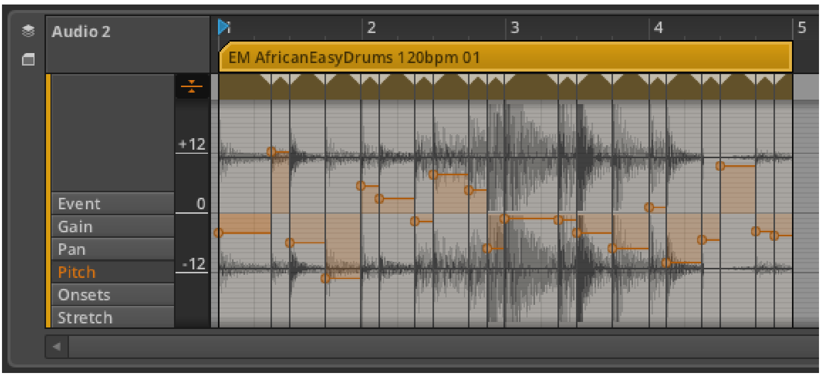


And now we have some variation in this expression.

You can see that the **Histogram** display now has some life in it. The horizontal positions are indicating the pitch values for various events — from -24 semitones on the left, to zero semitones in the middle (no pitch shift), to +24 semitones on the right. The vertical position of the chart roughly indicates the number of events found near that value.

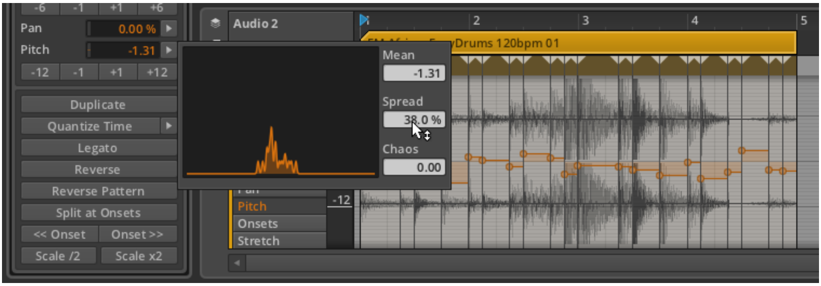
The distribution shown here is weighted toward the left (negative) side, and indeed, the *Mean* is telling us that *-1.31* semitones is the current average of all values. The **Inspector Panel** displays an identical *Pitch* value, showing that these two controls are identical.

The *Chaos* value is set in the units of the selected parameter, so it is *25.25* semitones of shift in this case. And because the pitch expression has a bipolar range, *25.25* semitones represents a distribution between -12.125 and +12.125 semitones.



Looking at the newly formed *Pitch* expression in the **Detail Editor Panel**, you can see that the highest point is right around +12 semitones (in the second audio event), and that the lowest point is right around -12 semitones (in the fourth event).

If we liked the shape of the expression but felt it was a little two extreme, we could call the **Histogram** back up and bring down the *Spread* value to narrow the overall range.

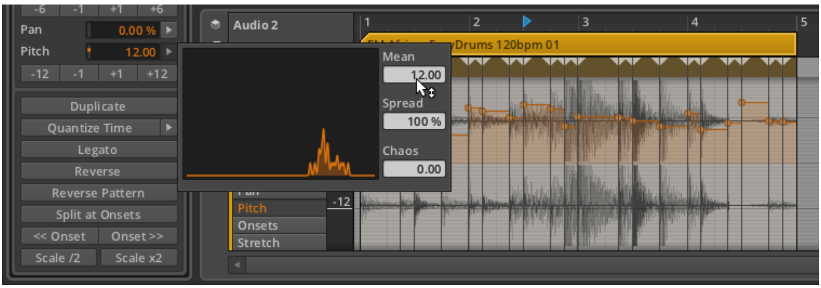


As the *Spread* value goes below *100%*, the range is indeed being shrunk, causing the histogram curve to become narrower and grow upwards — an indication that more of our 20 points are landing close to each other. But the shape of the curve is comparable to where it started.

Interestingly, the *Chaos* value was back at *0.00* when we brought the **Histogram** back up. Actually, this happened immediately after the *Chaos* setting was made and the mouse was released. And the same was true of the *Spread* function just now, as it will return to *100%* once you let go.

Each of these values represents an amount to change the current distribution of points. Unlike *Mean*, these values reflect only the future action and nothing about the present situation.

Finally, we can indeed use the *Mean* function to shift the whole expression so that zero is no longer near the center.



By moving the *Mean* to *12.00*, the average value is now a shift of one octave up with all variation landing just around that. (Again, we could have used the *Pitch* parameter to make the exact same adjustment.)

So that is a brief overview of how the **Histogram** works and an example of what you can do with it. We have spent this much time on it because the **Histogram** is available all across Bitwig Studio, whenever a group of numeric values can be selected together.

10. Working with Note Events

As we work with Bitwig Studio to assemble music, there are two forms of source material that we can use. One form is audio events, which was covered thoroughly in the last chapter. The other is *note events* — or simply *notes* — which we will investigate in this chapter.

As the introduction to the last chapter suggested, these two chapters are really parts one and two of working with the contents of clips. Accordingly, the format of this chapter is highly similar to the previous one, with many of the same issues and concerns being presented from the perspective of notes. And consistent with the rest of this document, ideas that reappear will reference the section where they were first discussed.

We will begin by revisiting the **Detail Editor Panel** to see how it works with note events, as well as the vast per-note modulation capabilities of Bitwig Studio. We then will see the last face of this panel as it allows us to work with multiple clips and tracks simultaneously. And after revisiting the **Inspector Panel** in the context of notes, we will take a look at the **Edit View**, the third and final panel set.

Let's sharpen our tools for working with that other type of musical content: *note events*.

10.1. The Detail Editor Panel, Note Clip Edition

The utility of the **Detail Editor Panel** should be clear by now, but the truth is that we have covered only half of it at best. We will start again with this panel because when it is focused on note clips, the same **Detail Editor Panel** adapts and provides slightly different options that are appropriate to the situation.

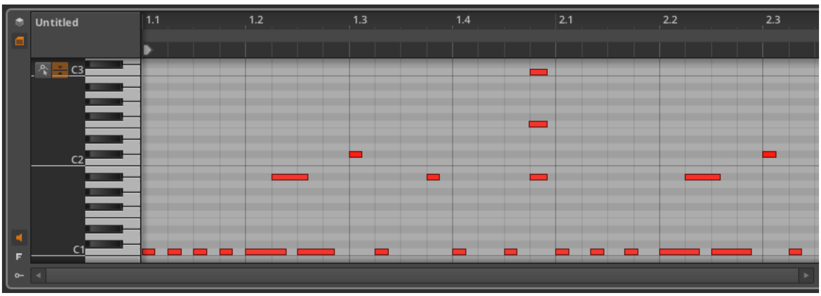
To better understand the incarnations of the **Detail Editor Panel**, let's take a moment to differentiate the structure of audio events and notes. (They are clearly made of different materials, but the way they are stored and structured are critical here.) The most important distinction is that while audio events are all of one kind, note events have pitches that allow us to distinguish them and make them overlap.

Only one audio event can occur at a time within a single clip, so while audio events can be arranged sequentially, they cannot be played simultaneously. And because no audio event has inherent priority over another, the last event placed in a certain position will "win."

If you move an audio event to a position already occupied by another event, the new event will effectively clear the position that it now occupies, leaving behind no trace of what was here.

This is because audio events cannot coexist. (Clips of all kinds behave in exactly the same fashion.) To illustrate this, moving the new event back to its original position will leave a hole where you had placed it.

The most important characteristic of each note is its *pitch*. This characteristic immediately gives us a way to distinguish notes from one another. And once we can distinguish notes by type, we can now have overlapping notes.

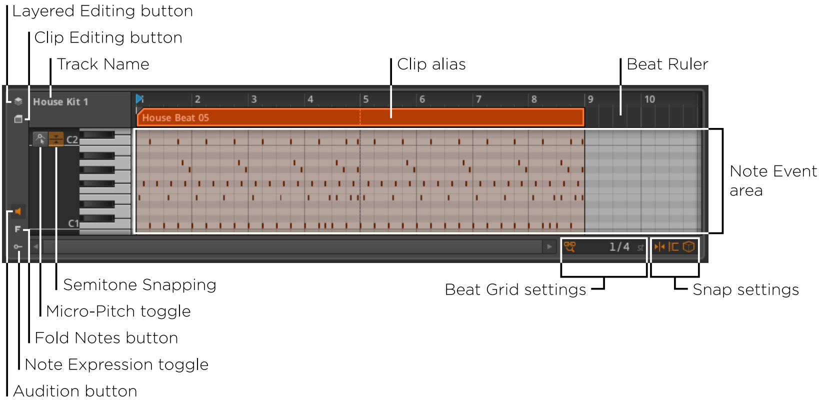


Chords and other overlapping gestures are a part of music, and note clips support them by allowing notes of different pitches to overlap. So while audio events are the smallest workable unit (and have their own headers to work with them), individual notes are the fundamental units here.

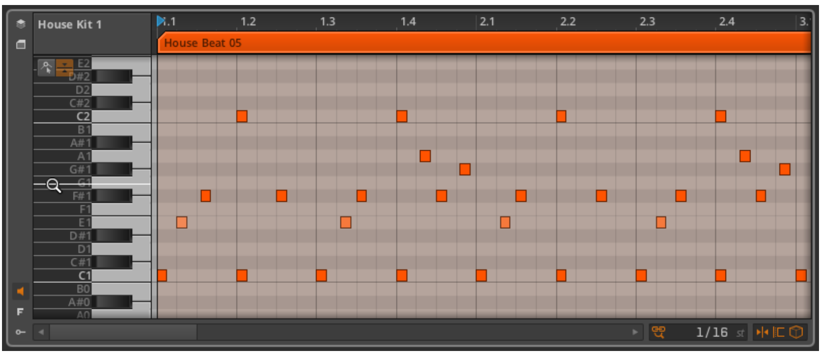
We will discuss the many similarities between how audio events and notes are edited. And they start in the **Detail Editor Panel**.

10.1.1. Layout of the Detail Editor Panel

Double-clicking a note clip in either the **Clip Launcher Panel** or the Arranger Timeline will call up the **Detail Editor Panel** and place its focus on that clip.



Much of this is familiar, such as the *Beat Ruler* (see section 3.1.1), the *clip aliases* (see section 8.2.1), the *Clip Editing button* (see section 8.2.2), this panel's own *beat grid settings* (see section 3.1.2), and the *snapping settings* (see section 4.2.2). The panel itself can still be vertically resized, but the y-axis can also be zoomed by clicking and dragging in the dark gray field just to the left of the piano keyboard.



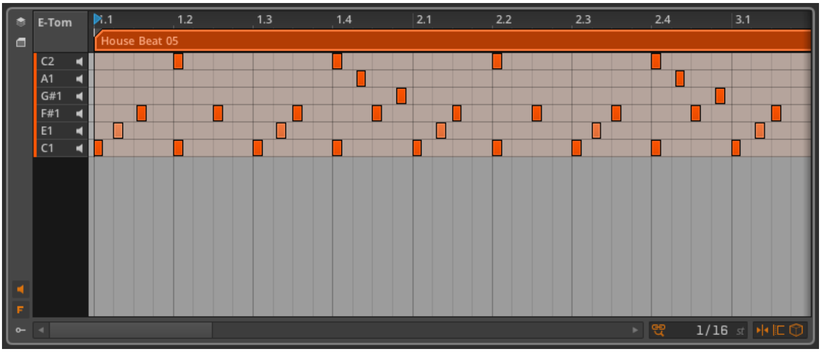
Three other new buttons have appeared in the bottom left corner of the **Detail Editor Panel**.

› When the *Audition button* is enabled, clicking and dragging any note to a new pitch will send a corresponding note to the track's device chain. This provides an audible preview of the action being considered.

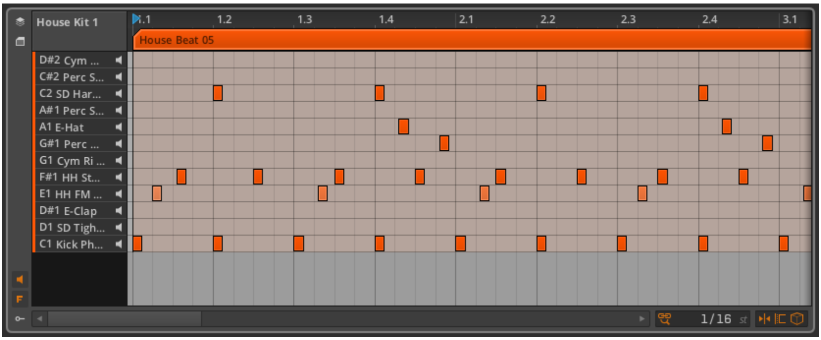
Additionally, clicking the piano keyboard to the left of the *note event area* will trigger a note when the Audition button is enabled.

› The *Fold Notes button* hides either unused or unavailable notes, depending on the instrument being used.

For nearly all instruments, only notes which are used on the current track (while in *track editing mode*) or for the current clip (while in *clip editing mode*) will be shown.

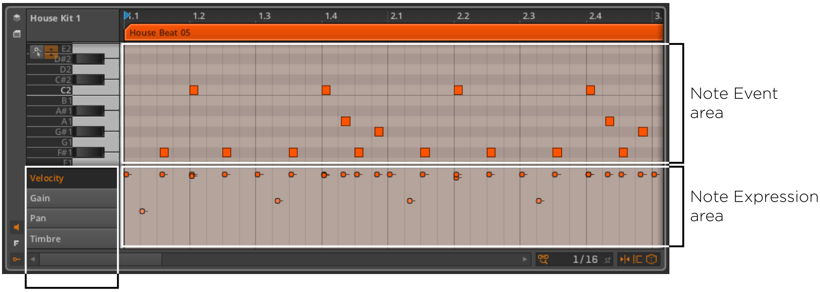


If the track's primary instrument is **Drum Machine**, then all notes which have available instruments will be shown.



In either case, everything else about the panel continues to work as usual.

› When the *Note Expression toggle* is enabled, the *Note Expression area* becomes visible below the Note Event area.



10.1.2. Note Event Expressions

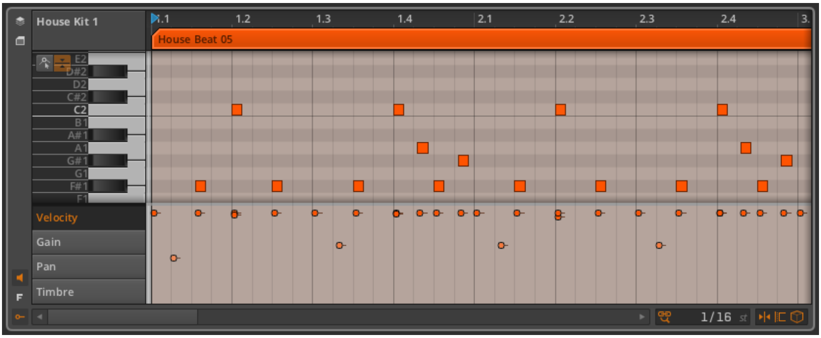
Like audio event expressions, *note expressions* are parameters that can be set for each individual note. Many of these parameters can change over the course of the note, making them like specialized automation curves.

Only one note expression can be focused on at a time, and you pick which expression to view by clicking its name in the list. We will take them from top to bottom.

> **Note**
>
> Of the available note expressions, only velocity will work with VST plug-ins. The others rely on Bitwig Studio's unique per-note modulation capabilities. These additional expressions will function properly only with Bitwig's instrument devices.

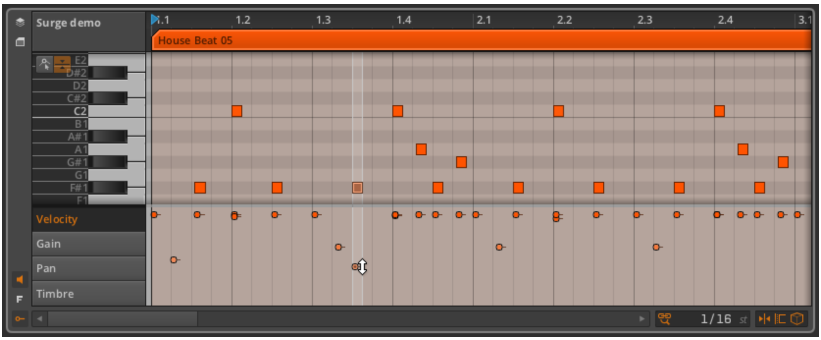
10.1.2.1. Velocity Expressions

Velocity expressions represent the strength with which each note should be triggered.



Similar to the MIDI specification, a *velocity expression* consists of a single value that is transmitted at the note's start. Each device determines how velocity will be used. With several Bitwig instrument devices, you can also use velocity expressions as modulation sources (see section 14.2.2).

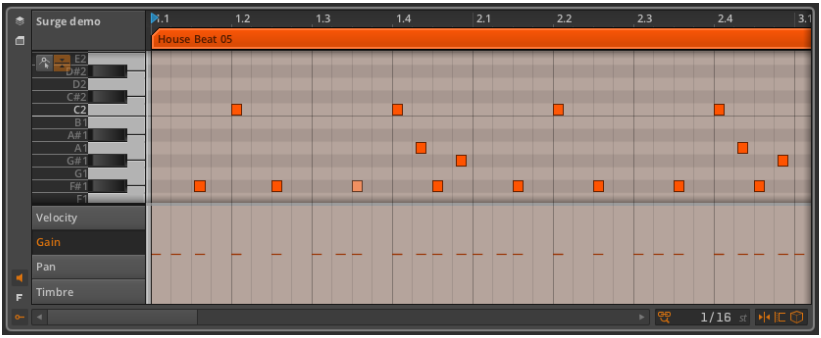
To adjust a velocity expression: mouse over the velocity expression so that a double-arrow cursor appears. Then click and drag the expression vertically.



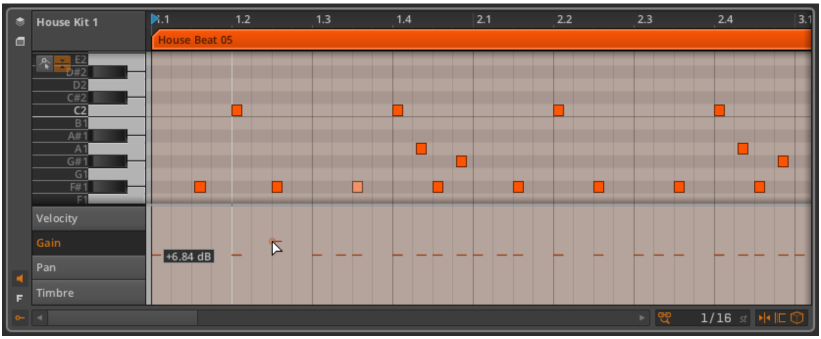
Notes are colored to match their clip's color, with the saturation of each note set relative to the strength of the note's velocity. A note at full velocity (*100%*) will be shown as the full color of the clip. As a velocity lowers, the color of that note will change.

10.1.2.2. Gain Expressions

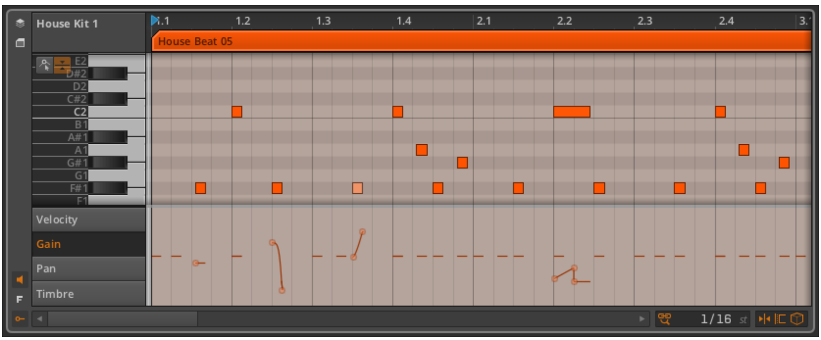
Gain expressions represent a level control for each note event.



To start with, each note's expression contains no individual points. By initially clicking and dragging an expression, you are both creating an initial point within the expression and defining the entire expression's value.



Once an initial point has been defined, additional expression points can be created and edited in the same way that automation points are (see section 8.1.2).

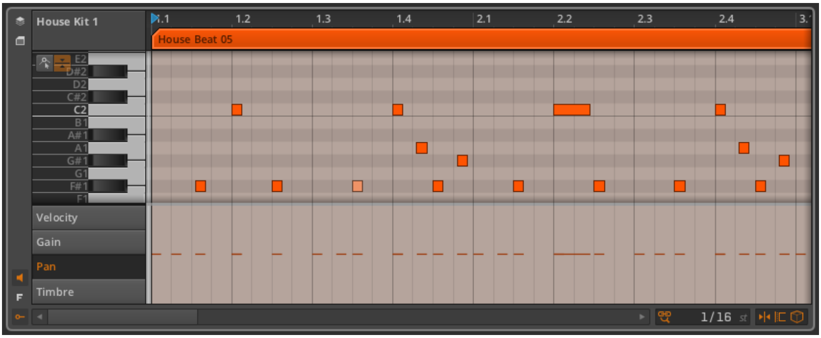


A *gain expression* is measured in units of decibels with the center line representing zero decibels of change (unity gain).

A gain expression is identical in function to volume automation. The difference is that the expression is applied at the beginning of the audio signal path — in this case, at the output of the instrument device that initially synthesizes audio signal. Volume automation is applied as the last stage of a track's signal flow (after the track's device chain and everything else).

10.1.2.3. Pan Expressions

Pan expressions represent a stereo placement control for each note event.



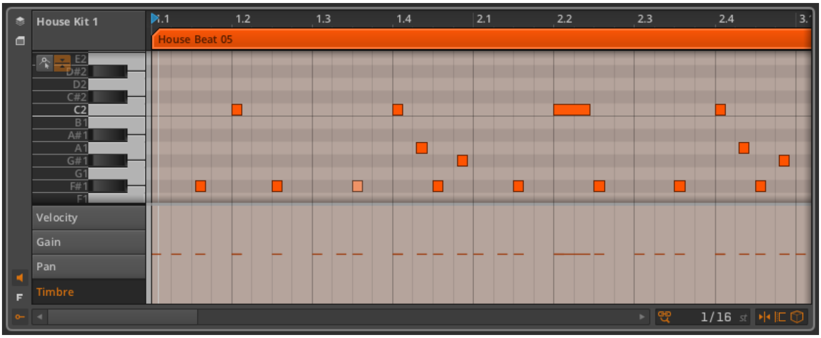
Once an initial point has been defined, additional expression points can be created and edited in the same way that automation points are (see section 8.1.2).

A pan expression is measured as a bipolar percentage with the center line at *0.00%* (center placement, or no panning adjustment), *100%* for hard right, and *-100%* for hard left.

As with the gain expression, the *pan expression* is applied at the beginning of the audio signal path. This has no direct interaction with *pan automation*, which is applied by the track mixer after the device chain.

10.1.2.4. Timbre Expressions

Timbre expressions represent an assignable modulation source for each note event.



Once an initial point has been defined, additional expression points can be created and edited in the same way that automation points are (see section 8.1.2).

The word *timbre* refers to a sound's tone color, but the timbre expression here has no fixed purpose. Rather, it can be used to freely modulate one or more parameters of the track's instrument device (see section 14.2). Mapping is done with the *TMB* modulation source, which is available on four Bitwig instruments: **FM-4**, **Organ**, **Sampler**, and **Polysynth** (see section 14.2.2).

A timbre expression is measured as a bipolar percentage with the center line at *0.00%* and the extremes at values of *100%* and *-100%*.

Similar to the gain and pan expressions, the *timbre expression* is applied within the instrument at the beginning of the audio signal path.

10.1.3. Micro-Pitch Editing Mode

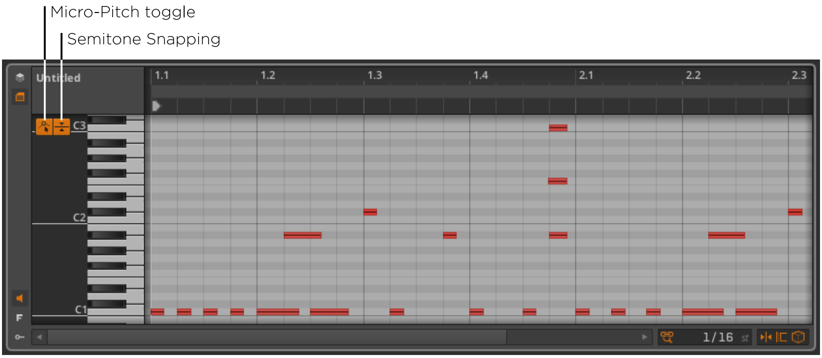
When working with notes, the **Detail Editor Panel** appears as a standard "piano roll" editor, with notes placed on their vertical pitch at the appropriate horizontal time. The notes can be created and edited in the exact same fashion as clips are (see section 4.2.1, section 4.2.2, and section 4.2.3).

By default, the **Detail Editor Panel** works with notes in the standard, discrete semitone fashion. But by enabling the *Micro-Pitch toggle*, we enter *Micro-Pitch editing mode*.

Note

Micro-Pitch editing relies on Bitwig Studio's unique per-note modulation capabilities. Micro-Pitch expressions will function properly with Bitwig's instrument devices, but not with VST plug-ins.

Micro-Pitch editing mode is not available while the Fold Notes button is enabled.



Thin lines are now drawn across the center of each note event. We can zoom in to make this easier to work with.

These lines are *Micro-Pitch expressions*. Like all other note expressions, Micro-Pitch expressions are per-note events, allowing the specific pitch of each note to be set precisely, or even to change the pitch of the note while it is played. You can think of Micro-Pitch expressions as a precise, polyphonic version of MIDI pitch bend, where each note played has its own pitch curve.

Micro-Pitch expressions are measured in semitones, with the center line at *0.00* (for no pitch shift), a maximum of *24.00* (two octaves up), and a minimum of *-24.00* (two octaves down).

Just a few examples of how this might be used:

› Building a chord with one of its notes bent while all others are held steady.

› Shaping a lead line with graceful transitions, where each note fades (perhaps with a gain expression) while gliding to the pitch where the next note will begin one.

› Carving out a solo, where the shape of the vibrato is precisely drawn.

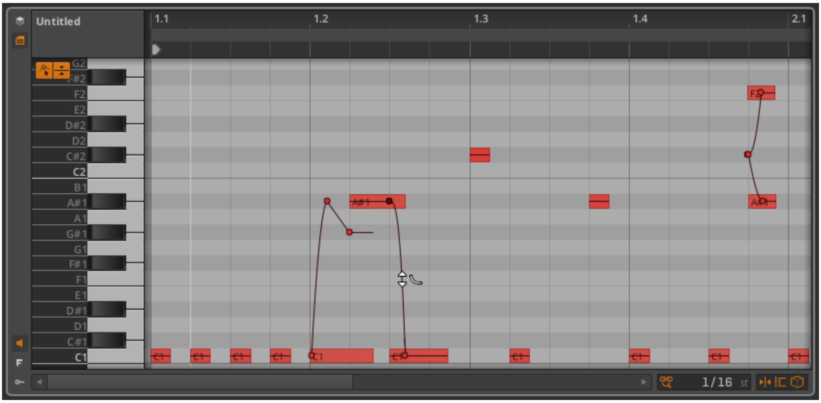
› Structuring a microtonal part, where each note's pitch is meticulously defined.

› Creating a part that combines any of these ideas, or something else altogether.

Like the other note expressions that can be automated, each Micro-Pitch expression is blank to begin with. The centered line represents that the note is tuned only by its standard pitch assignment.

By initially clicking and dragging the Micro-Pitch expression, you are both creating an initial point within the expression and defining the entire expression's value. In most cases, you will want to single-click the expression to start.

Once an initial point has been defined, additional Micro-Pitch expression points can be created and edited in the same way that automation points are (see section 8.1.2).



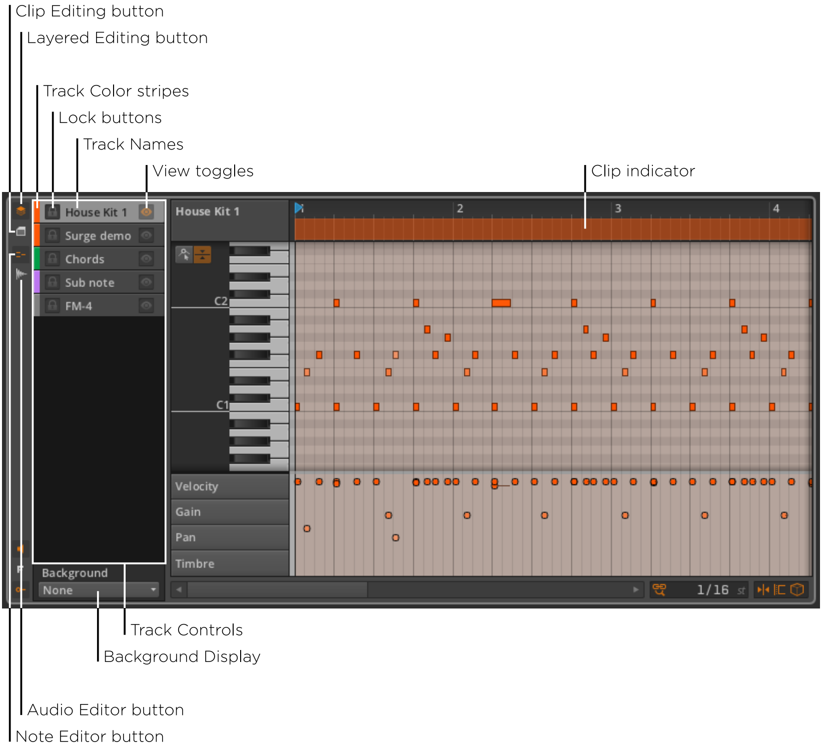
The *semitone snapping* option causes Micro-Pitch expression points to snap to whole number semitones. As with the position snapping options (see section 4.2.2), holding [SHIFT] will toggle this behavior. Semitone snapping is enabled by default.

10.1.4. Layered Editing Mode

We have seen the **Detail Editor Panel** work at various levels. We examined the panel while it focused on a single clip at a time in *clip editing mode*. We have also (and primarily) explored the panel while it focused on all contents of a track in *track editing mode*. And now there is one, larger level left to explore.

Layered editing mode still has a *clip editing button* for letting us toggle between clip or track editing mode. But once we have chosen that mode, entering layered editing mode allows us to view and edit several clips or tracks together. So once we pick the clip or track paradigm, we can then zoom out and work with several of those side by side.

We enter layered editing mode by enabling the *Layered Editing button*.



In the image above, we are in track editing mode. When we were previously in track editing mode within the **Detail Editor Panel**, the top of the panel displayed a *clip alias*. While track editing in layered editing mode, we now have a *clip indicator* instead. This indicator still shows us the start and end times of displayed clips, but the clip's name is no longer present and its length and position can no longer be manipulated.

Other than that, the right side of the panel is unchanged. The left side of the panel, however, contains several new items.

On the far left, top edge of the **Detail Editor Panel** are two buttons — the Layered Editing button and the Clip Editing button — which are already familiar. And if as in the image above the Clip Editing button is disabled, below it will be two new buttons that form a toggle pair.

If the *Note Editor button* is enabled, the **Detail Editor Panel** will focus on note containers as we have examined in this chapter. If the *Audio Editor button* is enabled, the **Detail Editor Panel** will focus on audio containers as we examined in the previous chapter. Only one of these can be enabled at a time so clicking either button toggles the current selection.

Taking all this together, we must select whether we want to use clip or track editing mode, and also choose whether we want to work with note or audio clips. For the current example, we will continue with note clips in track editing mode.

10.1.4.1. Layered Editing in Track Mode

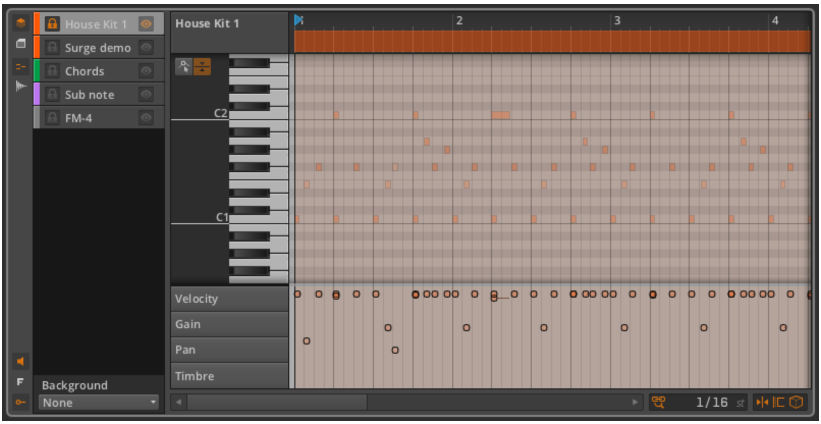
Now that our modes are set, the resizable *track controls* section houses a set of controls for each instrument and hybrid track in the current project. These controls include:

› *Track Color stripe*: A swatch of the track's assigned color.

This can be helpful when viewing multiple tracks as each track's notes are still tinted their track's color.

› *Lock button*: When enabled, protects the track's data from being selected or altered.

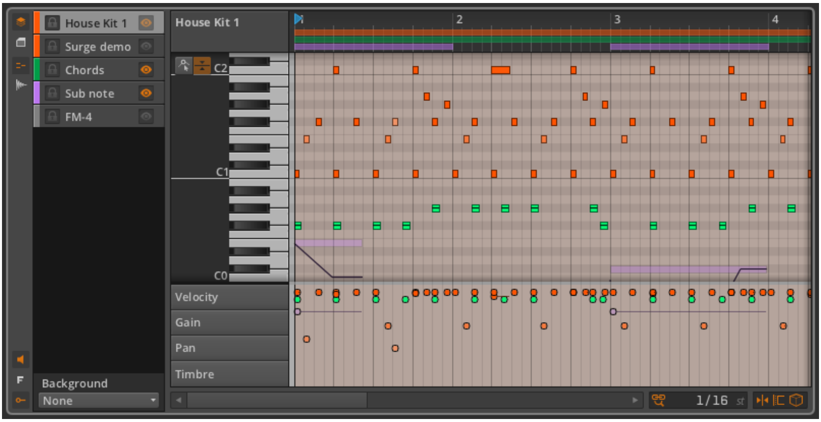
When a locked track is visible, its contents are still shown but significantly dimmed.



› *Track Name*: The title assigned to the track.

› *View toggle*: Toggles the track's visibility.

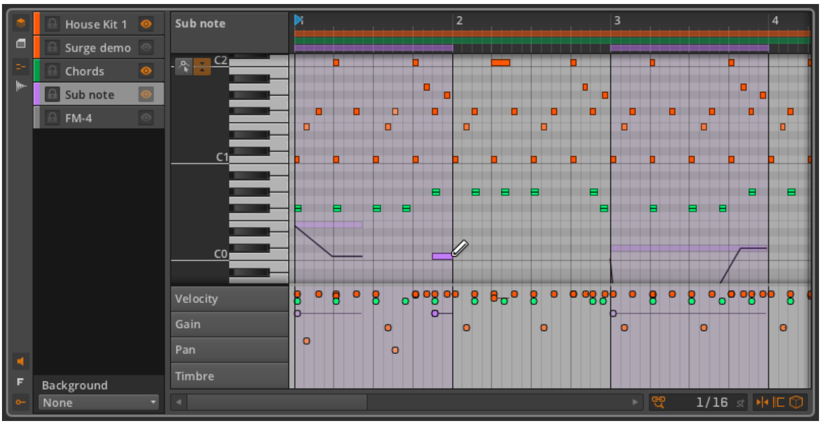
To make a track visible: enable its view toggle.



All aspects of unlocked visible tracks are editable with the techniques we have seen. Data from various tracks can also be edited together in this fashion, and objects can even be placed in relation to one another with *object snapping* (see section 4.2.2).

While editing, you may want to create new notes either by drawing them in with the pencil tool, by pasting them, or by some other method. Since these functions can be applied to only one track at a time, you can designate one of the visible tracks as a *target track*.

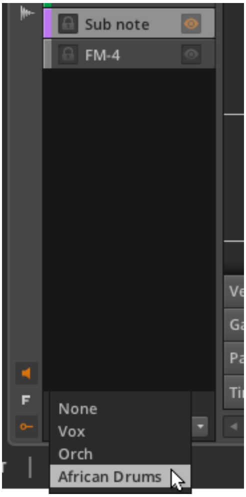
To select the "target" track: click the track's name to highlight it.

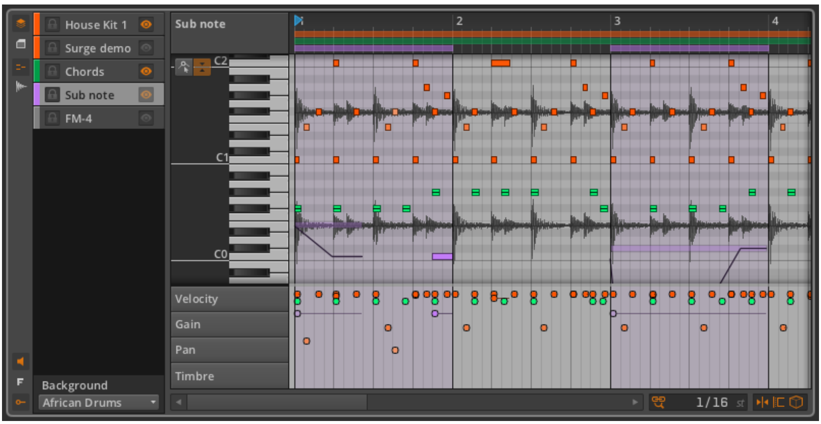


If the track you select was not previously visible, it will be made visible while selected.

Any clip indicators for the target track will also shade the note event area to indicate both the boundaries you are working within and how those boundaries might change by moving notes into empty space.

While in the Note Editor, the *background display* setting is the final interface item. The menu labeled *Background* appears below the track controls and allows you to pick a background for display behind the note event area. The choices are either *None* (for no background) or any of the audio or hybrid tracks in the current project.

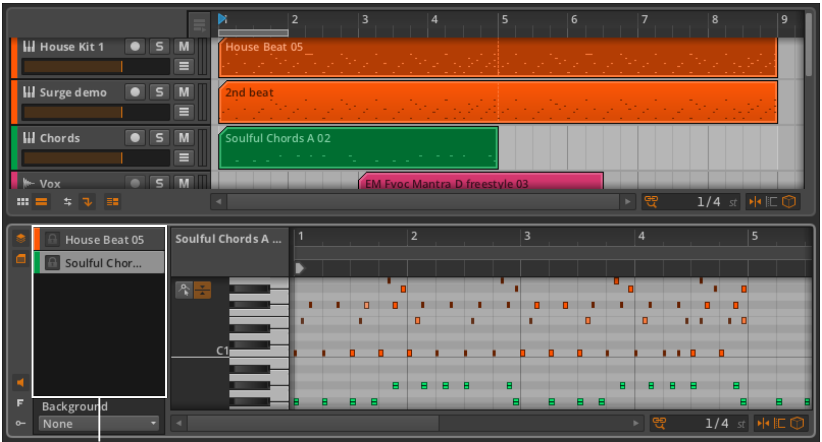




This setting is purely visual but can serve as a helpful reference.

10.1.4.2. Layered Editing in Clip Mode

Switching from track editing mode to clip editing mode presents a few structural differences.



Clip Controls

Again, the right side of the **Detail Editor Panel** is largely unchanged from its standard clip editing mode layout.

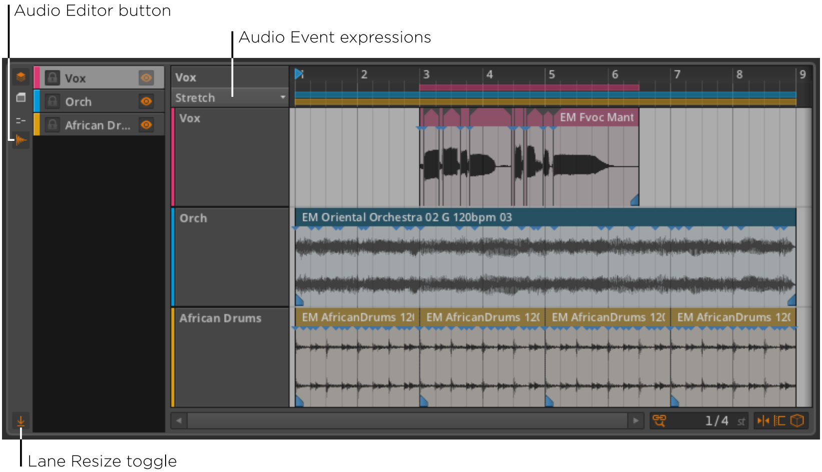
On the left side of the panel, the track controls have been replaced by *clip controls*. The primary difference here is that only clips which are currently selected in the active sequencer (either the Arranger Timeline or the **Clip Launcher Panel**) will be shown as options.

Because your selection is made in the sequencer, no view toggles are needed. Also the Note Editor and Audio Editor buttons will appear only when both clip types are selected.

Otherwise, this configuration works as expected.

10.1.4.3. Layered Editing with the Audio Editor

Switching from the Note Editor to the Audio Editor also presents a few structural differences.



In track editing mode, audio events can be freely worked with as described in the previous chapter. In clip editing mode, both audio events and clips can be worked with.

Audio expressions can also be worked with in both modes. A single *audio event expression* menu appears above the track headers to determine which expression is globally displayed.

And again, events and/or expressions can even be set in relation to one another with *object snapping* (see section 4.2.2).

The last new interface option is the *Lane Resize toggle*. When enabled, resizing the **Detail Editor Panel** also tries to resize each individual track/clip lane in order to fit the available space.

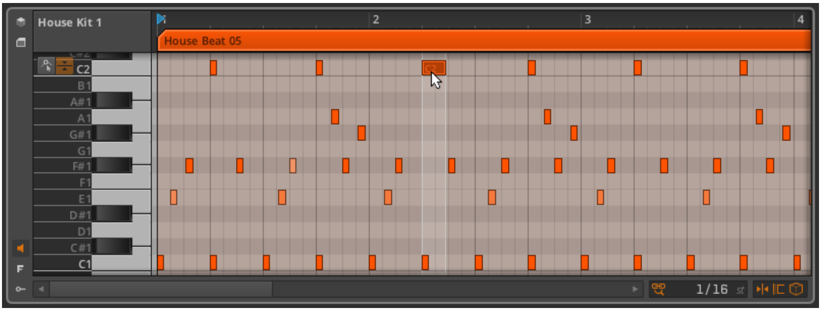
Otherwise, this editor works as expected.

10.2. Inspecting Note Clips

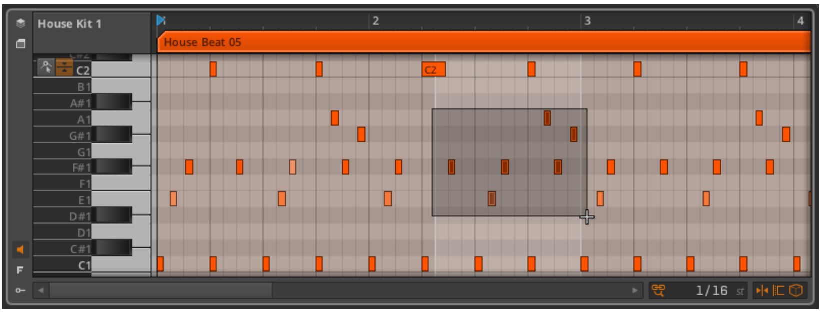
As with audio events, the **Inspector Panel** is a critical way to both access the details of note events and edit them most effectively. To focus the **Inspector Panel** on notes, we must first select them within the **Detail Editor Panel**.

10.2.1. Selecting Notes

To select a single note: single-click it.



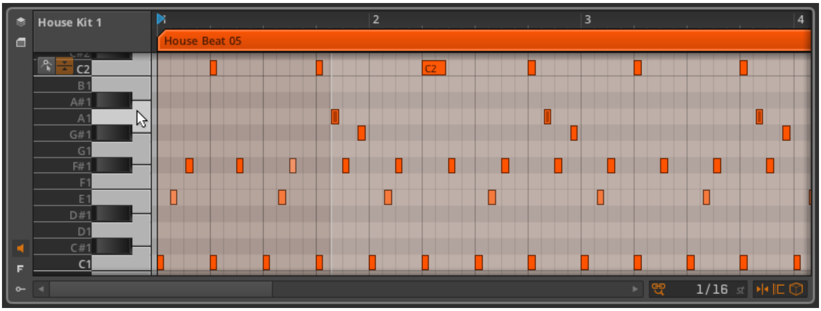
To select multiple notes: click a blank area and drag a rectangle around the desired notes.



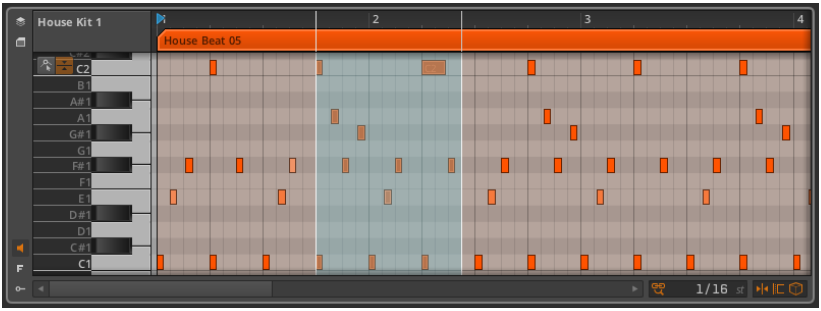
Other ways to select multiple notes include:

› After selecting one note, [CTRL]-click ([CMD]-click on Mac) additional notes to grow the selection.

› Click a note on the piano keyboard to select all displayed notes of that pitch.



› With the Time Selection tool, click and drag over the time area for which all displayed notes should be selected.



(To normally click and drag the notes after they are selected in this way, you can switch back to the Object Selection tool.)

To select the next note: press [ALT]+[RIGHT ARROW].

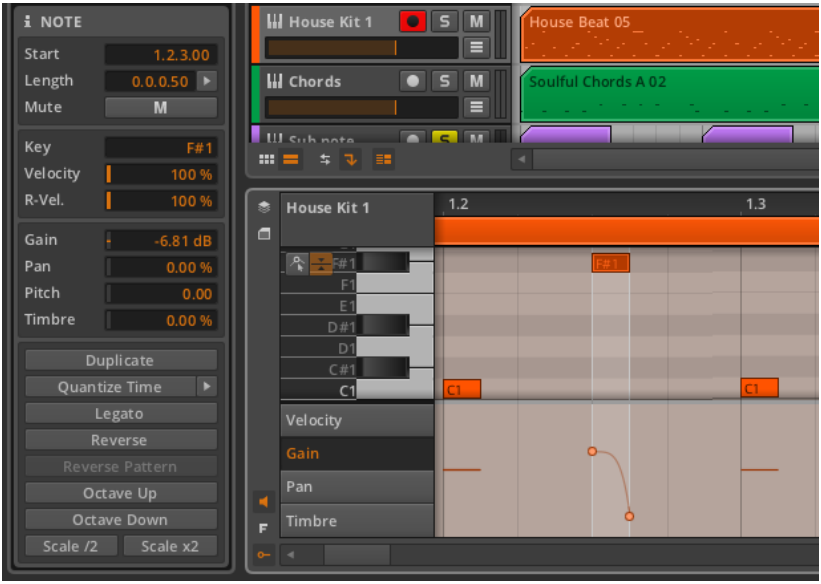
To select the previous note: press [ALT]+[LEFT ARROW].

If you have one note selected, you can similarly grow the selection by pressing [SHIFT]+[ALT]+[RIGHT ARROW] or [SHIFT]+[ALT]+[LEFT ARROW].

Once a note selection is made, the **Inspector Panel** will display relevant settings and functions.

10.2.2. The Inspector Panel on Note Events

As with audio clips and events, selecting a note clip makes certain parameters and functions available in the *NOTE* section of the **Inspector Panel**, but by selecting a note event itself, the **Inspector Panel** provides all settings and functions relevant for the selected event(s).

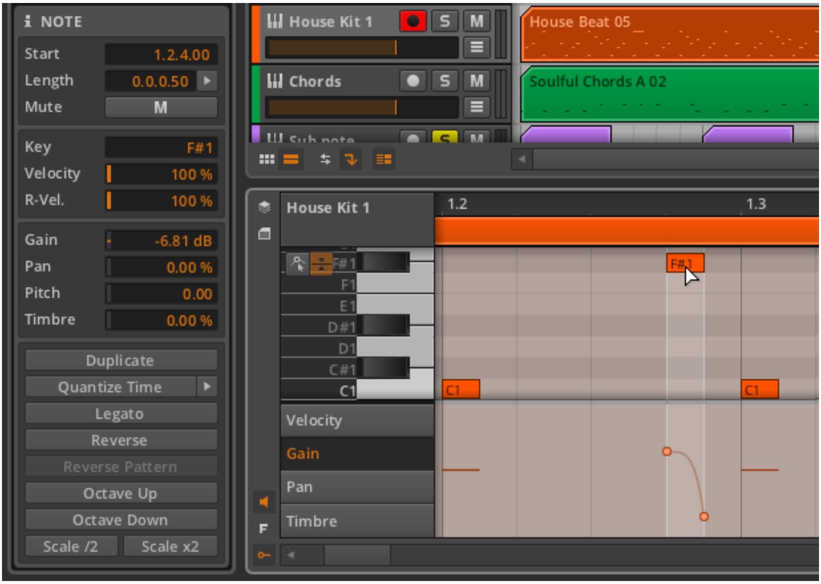


We will take these one section at a time.

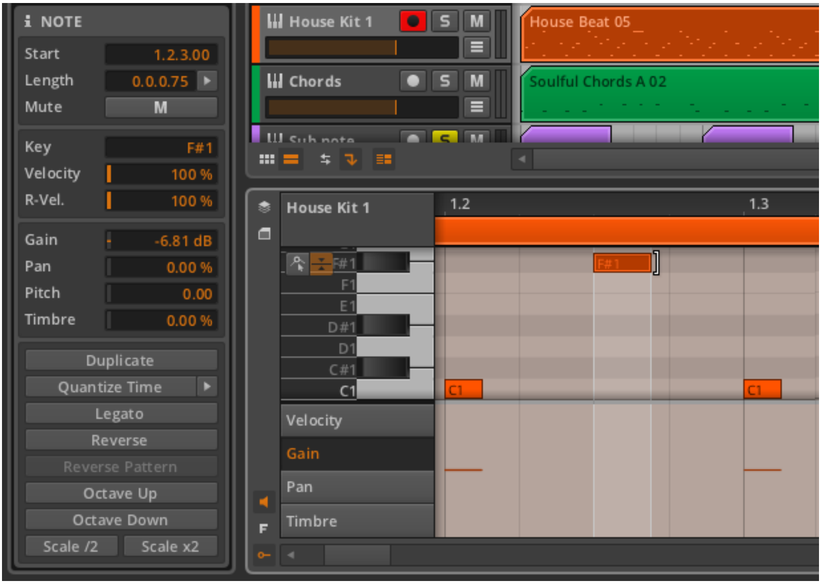
10.2.2.1. Timing and Mute Section

These settings relate to the musical position of the selected note and whether it is muted:

› *Start* sets the start position of the event within its parent clip or track. Adjusting this position will move the note event as it exists, the same as clicking and dragging the event within the **Detail Editor Panel**.



› *Length* sets the duration of the event within its parent clip. Adjusting this duration will simply lengthen or shorten the note event, the same as using the bracket cursor to adjust the right edge of the note.



› *Mute* toggles whether or not the event is disabled on playback.

10.2.2.2. Note Properties Section

These parameters are attached to each selected note's beginning and ending:

› *Key* sets the root pitch that the note is set to. This is shown as a MIDI note value, where *C3* is roughly 261.262 Hertz ("middle C") and *A4* is 440 Hertz. Adjusting this value is the same as moving the note higher or lower.

Any Micro-Pitch expressions are applied relative to the note's *Key* setting.

› *Velocity* sets the strength with which the note should be initially triggered. It is set on a scale from *0.00%* to *100%*, and this is just another representation of the note's velocity expression (see section 10.1.2.1).

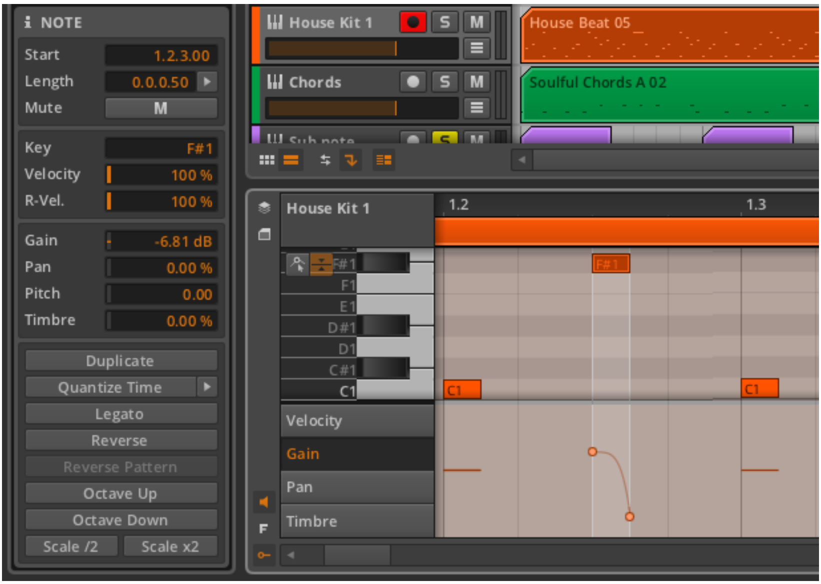
› *R-Vel.* stands for *release velocity*, and it sets the speed with which the note should be released. It is set on a scale from *0.00%* to *100%*. This parameter is implemented in whatever way the instrument device desires.

10.2.2.3. Expressions Section

This section exposes four of the expressions we have covered: *Gain* (see section 10.1.2.2), *Pan* (see section 10.1.2.3), *Pitch* (also known as the Micro-Pitch expression; see section 10.1.3), and *Timbre* (see section 10.1.2.4). While these expressions have completely different functions, they are programmed in the same fashion.

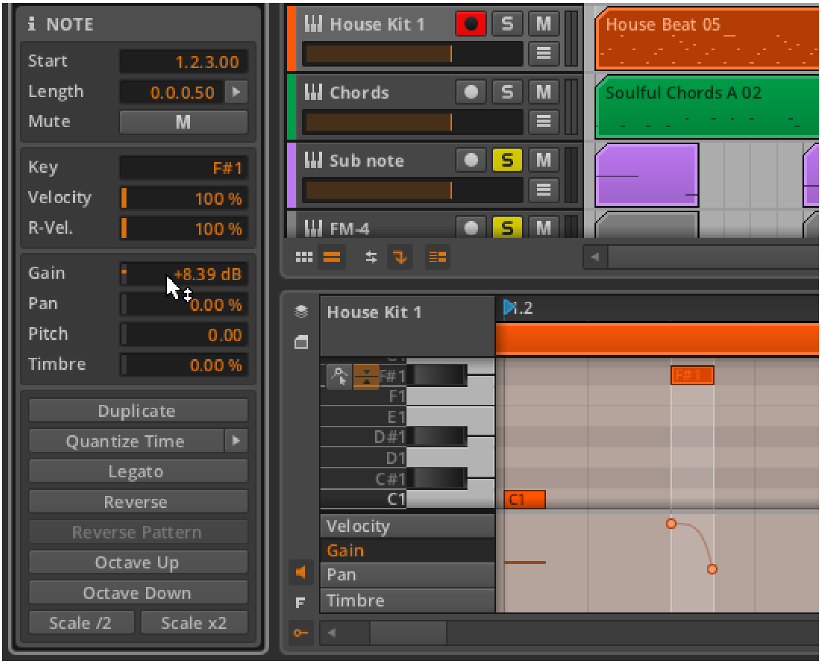
Most of these expressions have their units defined, with *Gain* set in decibels, and both *Pan* and *Timbre* set with bipolar percentages. The unlabeled *Pitch* is set in semitones, indicating the relative shift.

These are all automation-type expressions, so each is able to be defined by a curve made of several values. Because of this possibility, each value in this section of the **Inspector Panel** actually represents the average of points in that expression. We can see this in action with the *Gain* setting.

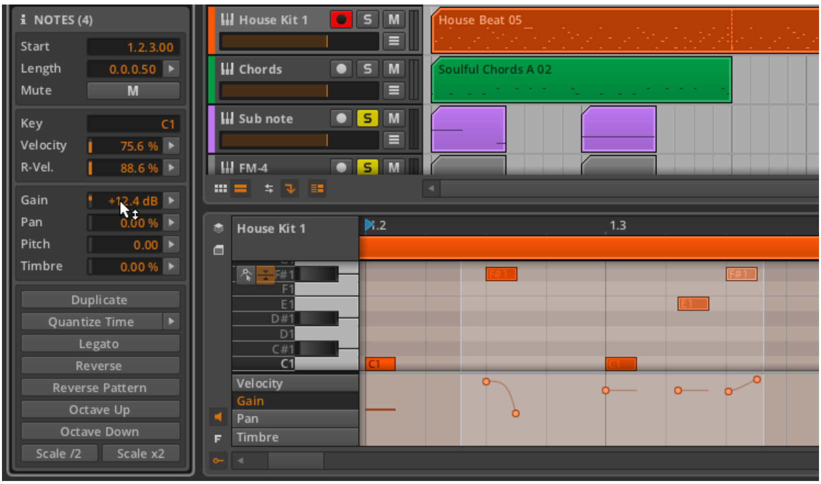


This note has a gain expression consisting of two points and a curve. The *-6.81 dB* listed for the *Gain* parameter is an average of these two points.

To adjust a note expression curve: change its listed average value.



This would work similarly if multiple note events were selected.

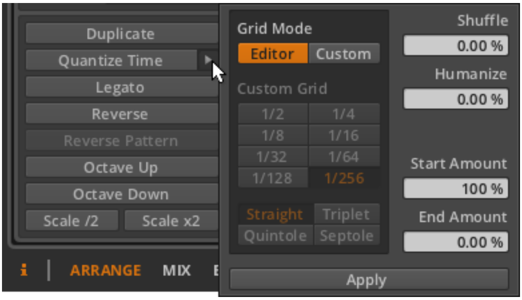


10.2.2.4. Function Buttons Section

These buttons execute the specified function on the selected note event(s):

› *Duplicate* places an exact copy of the selected event immediately after it. This function is also available from *Edit* › *Duplicate Note(s)* or by pressing [CTRL]+[D] ([CMD]+[D] on Mac).

› *Quantize Time* moves the start and/or end times of the selected note in relation to a beat grid. The parameter pane for this function appears when the right-arrow button is clicked.



› *Grid Mode*: Determines whether to adopt the grid settings from the current *Editor* or to allow *Custom* grid settings.

› *Custom Grid*: Exclusive *beat grid resolution* and *beat grid subdivision* settings (see section 3.1.2) for the quantize function.

> **Note**
>
> This is available only when *Grid Mode* is set to *Custom*.

› *Shuffle*: Amount of swing/groove (see section 2.1.1) applied to the beat grid for the quantization function.

› *Humanize*: Amount of randomness added to the quantize function, with the intention of mimicking human imperfection.

› *Start Amount*: Amount of quantization applied to each selected event's start position.

For example, a setting of *50.0%* would move a selected event's start position halfway to the closest grid point. A setting of *100%* places the event exactly on the closest grid point.

› *End Amount*: Amount of quantization applied to each selected event's end position.

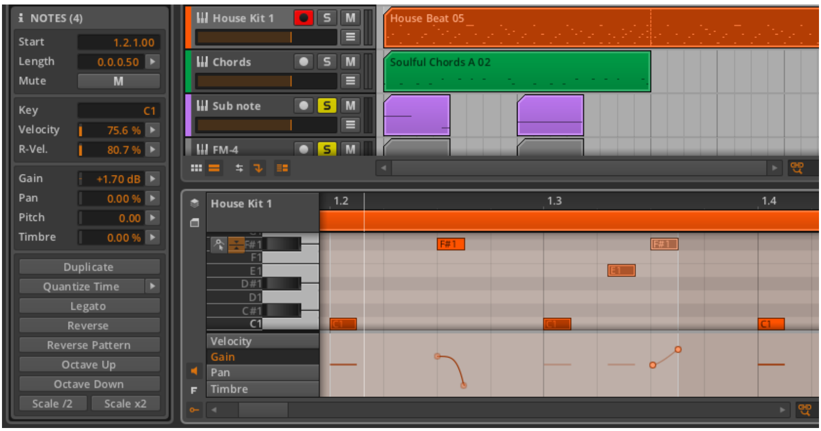
Note

Humanize is the last factor applied in the quantize function. So a *Start Amount* of *100%* might not place events directly on the grid if *Humanize* is enabled.

The quantize function can be executed by either clicking the *Apply* button at the bottom of the parameter pane, or by clicking the *Quantize Time* button itself.

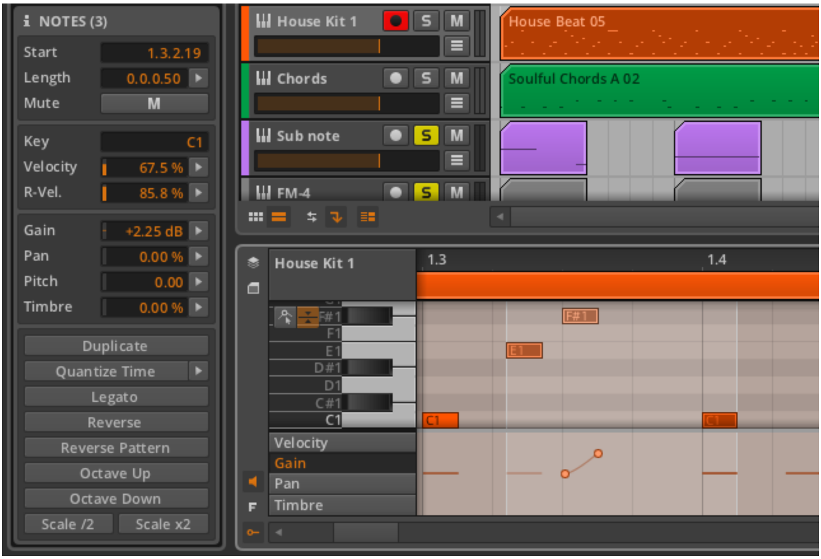
› *Legato* adjusts the length of each selected note event so that it ends immediately before the next event begins, creating a continuous series of events.

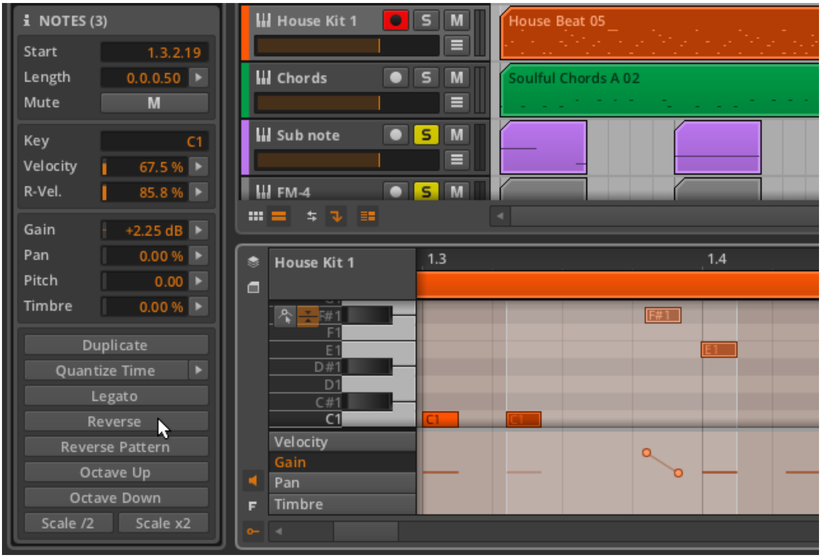
The following images demonstrate a group of selected events both before and after the *Legato* function is applied:



› *Reverse* flips the selected event around, causing it to play backwards.

The following images demonstrate a group of selected events both before and after the *Reverse* function is applied:





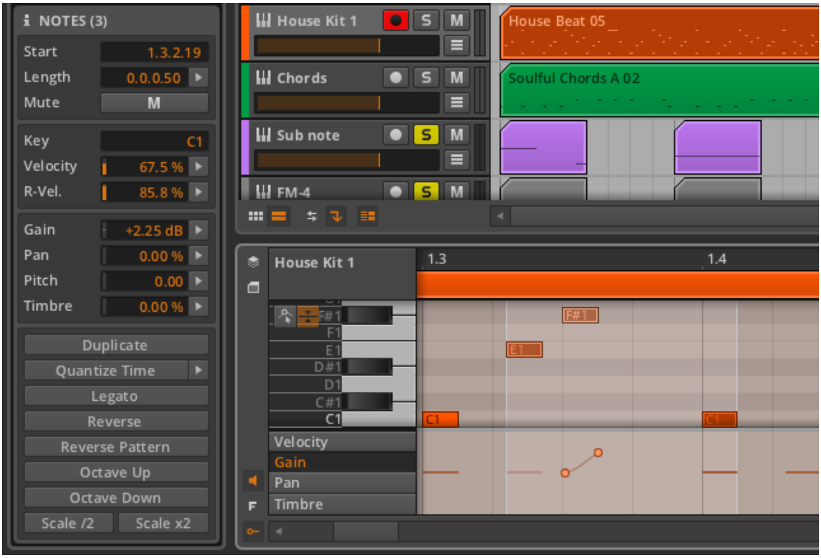
Notice that the expressions are also reversed.

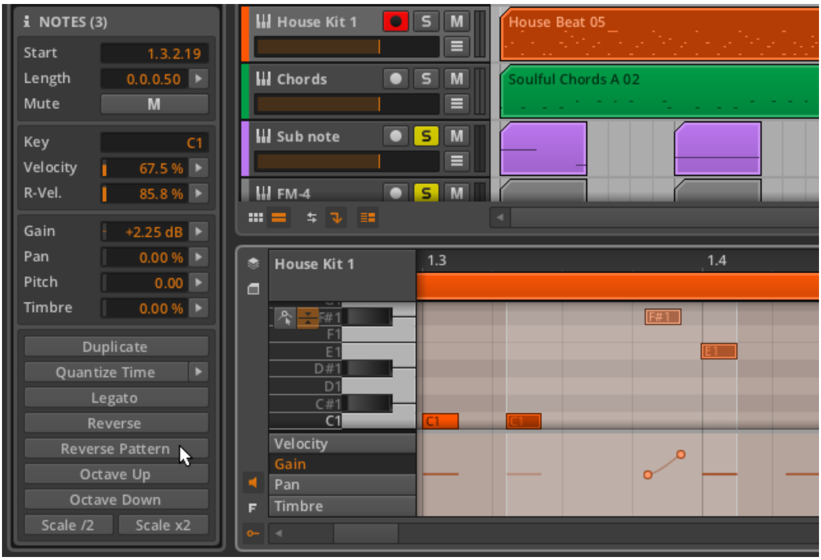
› *Reverse Pattern* flips the order of a group of selected events. This does not cause each event and its expressions to play backwards, but rather causes the last event to be played first, etc.

Note

This function will work only when multiple events are selected.

The following images demonstrate a group of selected events both before and after the *Reverse Pattern* function is applied:





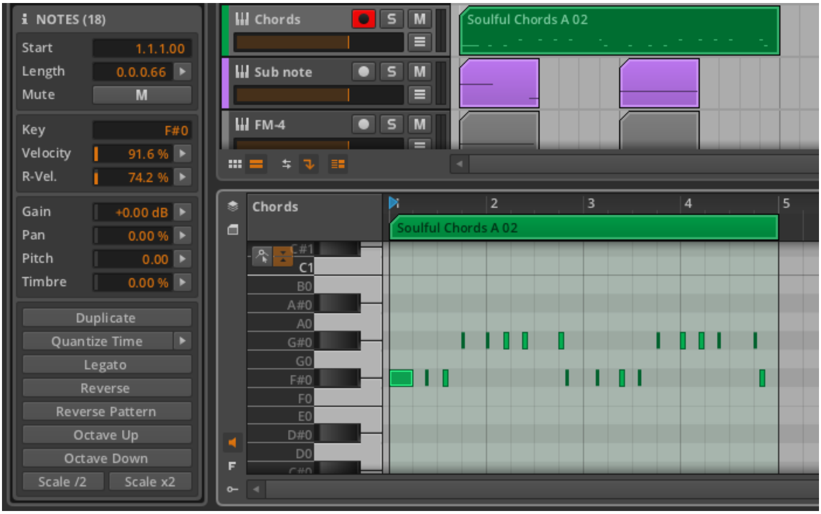
Notice that the expressions are preserved.

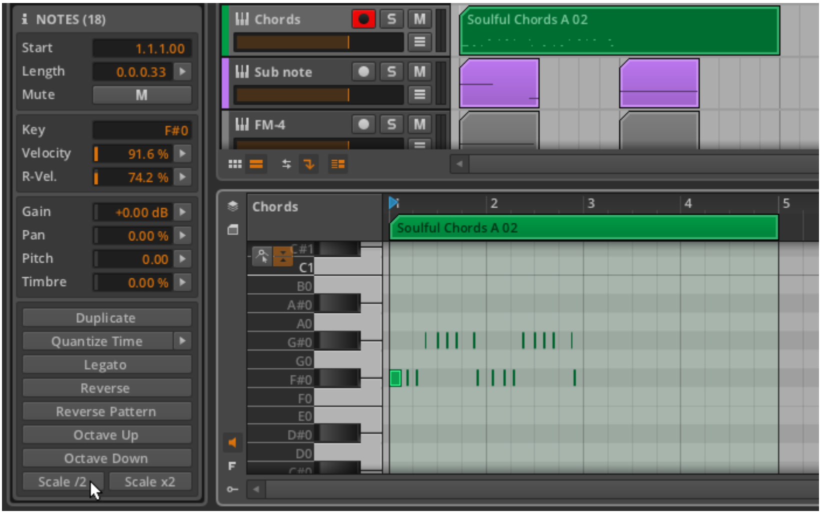
› *Octave Up* slides the selected event up 12 semitones (in musical notation, *8va*). This function is also available by pressing [SHIFT]+[UP ARROW].

› *Octave Down* slides the selected event down 12 semitones (in musical notation, *8vb*). This function is also available by pressing [SHIFT]+[DOWN ARROW].

› *Scale /2* halves the length of the selected event, effectively causing it to play back twice as fast. All expressions are also proportionally adjusted.

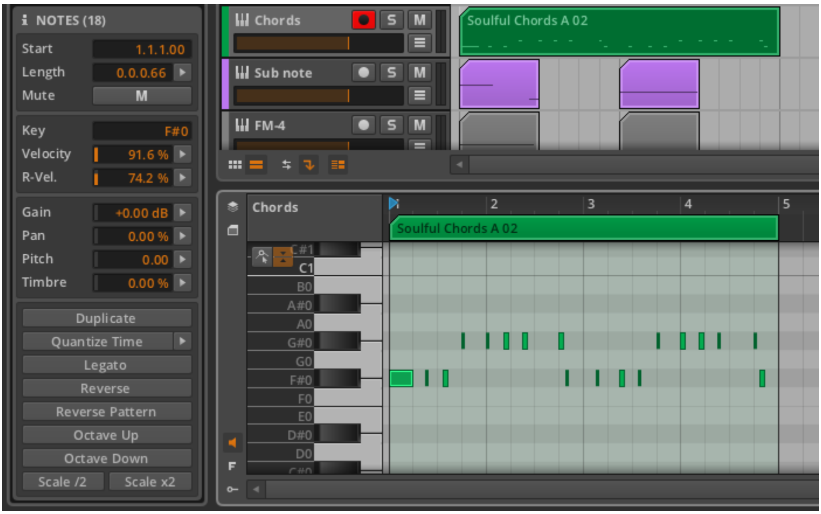
The following images demonstrate selected note events both before and after the *Scale /2* function is applied:

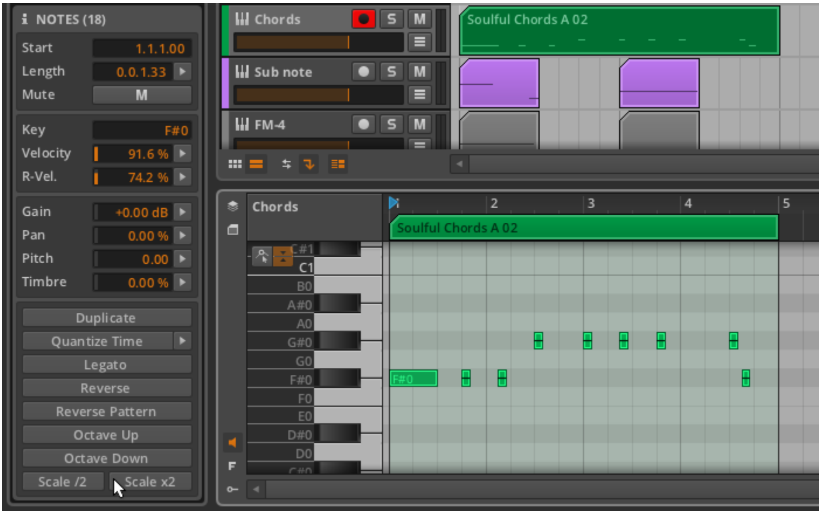




› *Scale *2* doubles the length of the selected event, effectively causing it to play back half as fast. All expressions are also proportionally adjusted.

The following images demonstrate selected note events both before and after the *Scale *2* function is applied:



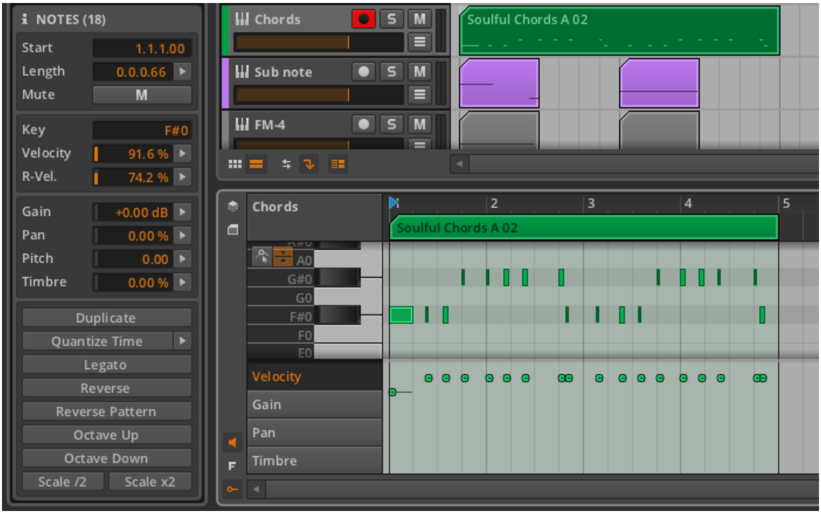


Note

Remember that events must fit within their parent clip.

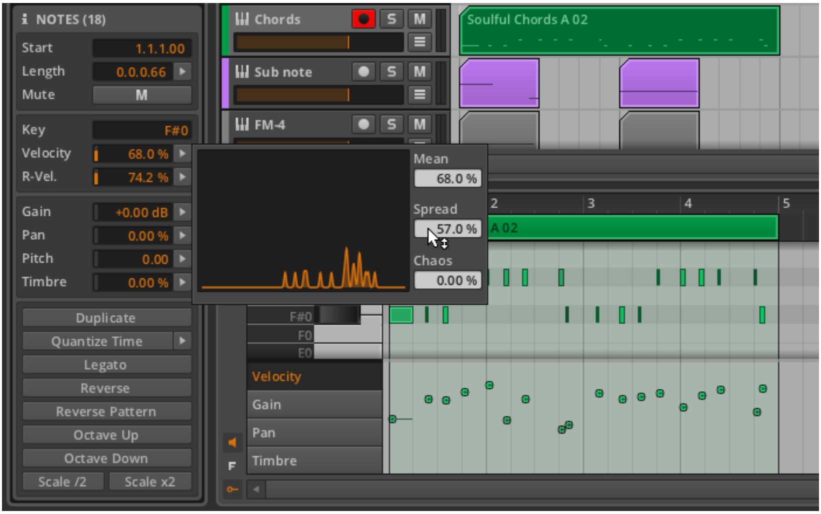
10.2.3. Working with Multiple Note Events

As it was with audio events, the **Histogram** becomes available when multiple note events are selected (see section 9.2.2.2).



In this example image, the **Inspector Panel** has labeled its bottom section as *NOTES (18)*, indicating that 18 notes are currently selected. And with this selection of multiple note events, the *Velocity*, *R-Vel.*, *Gain*, *Pan*, *Pitch*, and *Timbre* parameters all can now use the **Histogram** interface for editing.

The **Histogram** works exactly the same as it did in the audio event context (again, see section 9.2.2.2). The **Histogram** can be useful in the note context, for example, when notes were programmed without much diversity in their velocities.



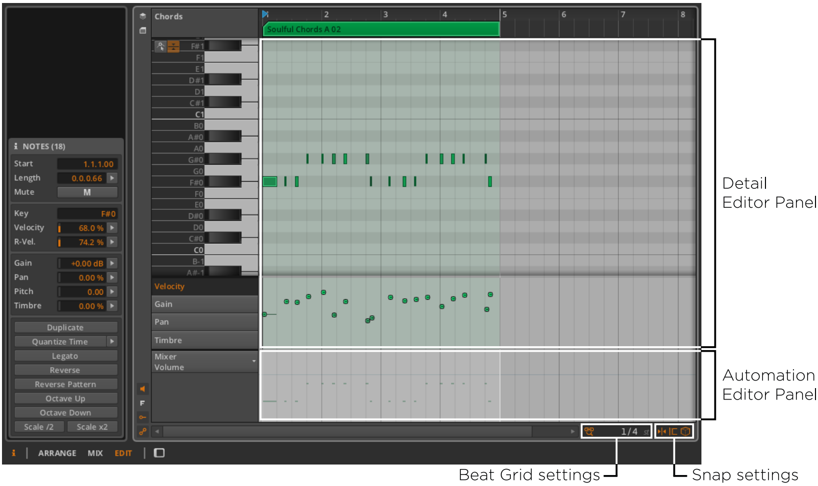
It doesn't take much to add subtle — or less subtle — variety with the **Histogram**. If you look, you will find places where it can aid your workflow.

10.3. The Edit View

Now that we have exhaustively covered both the **Automation Editor Panel** (in chapter 8) and the **Detail Editor Panel** (in both this chapter and chapter 9), we can now take a look at the **Edit View**, the last of Bitwig Studio's three views.

As we've discussed before, each view is a curated layout of Bitwig Studio's panels that is meant to serve a particular musical task. The **Arrange View** is purposed for assembling music, placing the important **Arranger Timeline Panel** centrally and giving you access to all panels around it. The **Mix View** centers around the **Mixer Panel**, focusing on the mixing board capabilities of each track while also streamlining the **Clip Launcher Panel** to facilitate improvisation.

Both of these views are oriented to show your project's tracks side by side, letting you craft a balance between them. But the **Edit View** is focused on the details of single tracks and clips.



The description just given and the image above should both be familiar at this point. The **Edit View** has two central panels: the **Detail Editor Panel** with an optional **Automation Editor Panel** fused beneath it. Aside from their positioning and the *Automation Editor Panel view toggle*, these panels work exactly as we have already learned them.

This combination allows you to focus on either the track or clip level so that you can work with the note/audio events contained there, the attached expressions, and the automation all beside each other. And putting the **Detail Editor Panel** front and center gives you much more display space for seeing more notes at a time — or, in the case of layered editing mode, more tracks. These are all welcome additions to the toolbox.

As a final point, the **Edit View** also strengthens the utility of *display profiles*. Since these profiles are meant to enable you at various stages of music production, you can probably imagine situations for having the full project on one screen (the "big picture") so that you can select a single clip or track and have its contents presented on the second screen (the "close up"). Again, once you scratch the surface, you will find uses for these functions in your workflow.

11. Going Between Notes and Audio

The previous two chapters dealt extensively with audio events and note events, which you could think of as our primary states of musical matter. And those last two chapters are the longest in this document because there is quite a lot to do in Bitwig Studio with audio and note events.

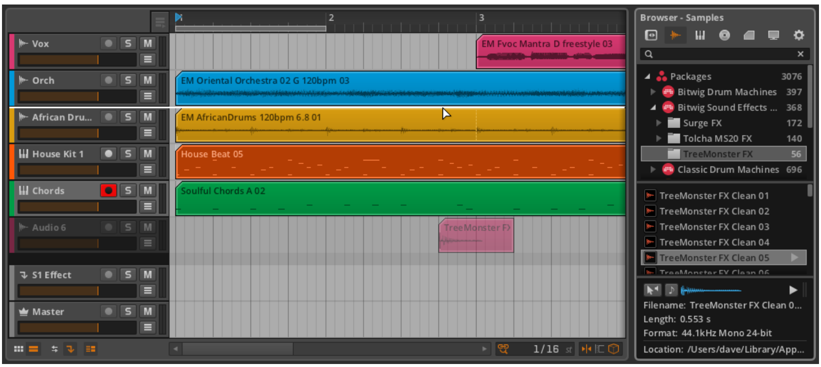
Working with notes has inherent advantages, as does working with audio. Without trying to tell you how either of these types "should" be used, it is fair to say that notes allow for a greater level of flexibility and control, and audio is both highly portable and can be wonderfully mangled.

But occasionally, matter changes state. This happens in the physical world when water freezes, and it also happens when you use Bitwig Studio to bounce a note clip in place. And just as ice sometimes melts, even audio can be sliced into note events.

This chapter will explore ways of taking audio materials into the note domain, ways of transforming note events into audio ones, and places where both coexist. We may not be altering nature here, but these options will only afford you more opportunities to customize your workflow and sound.

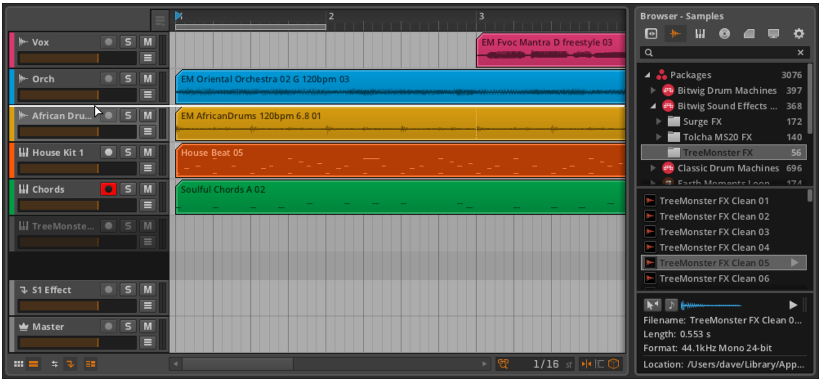
11.1. Loading Audio into a New Sampler

We learned early on how to import media files from the **Browser Panel** as clips. We saw how to bring clips of any kind into the **Arranger Timeline Panel** (see section 4.2.1) and into the **Clip Launcher Panel** (see section 5.2.1). In both cases, we also saw how to create a new track for that clip by dragging it to the space between any two existing tracks.

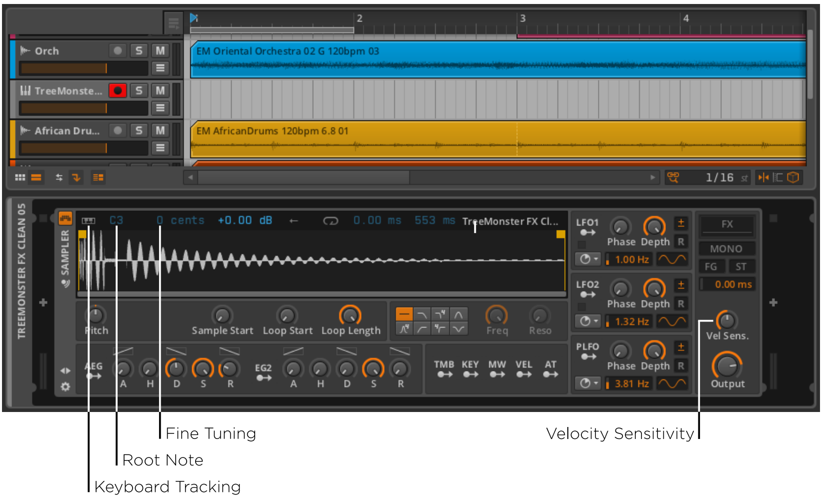


When importing an audio file, Bitwig Studio provides one additional option worth mentioning here.

To load an audio file into a new Sampler device on a new instrument track: click and drag the clip from the **Browser Panel** to the space between two existing track headers.



Once the mouse is released, a new instrument track will be inserted in that place, and the track will be selected.



With the track record enabled, you can now use notes to trigger the audio that was just loaded.

Rather than exploring the **Sampler** in any great detail, we'll look at just a few parameters that affect how the notes you play are interpreted by **Sampler**.

› *Keyboard Tracking*: When disabled, any note triggers the sample at its original pitch. When enabled, each note's pitch setting will change the playback speed and pitch of the sample.

› *Root Note*: The note which will play the sample at its original pitch. This setting takes effect only when *Keyboard Tracking* is enabled.

› *Fine Tuning*: A small interval adjustment for the *Root Note* setting, in units of cents (hundredths of a semitone). This setting takes effect only when *Keyboard Tracking* is enabled.

› *Velocity Sensitivity*: The amount that each note's velocity affects the loudness of the sample. At the lowest setting (*+0.00 dB*), velocity is ignored.

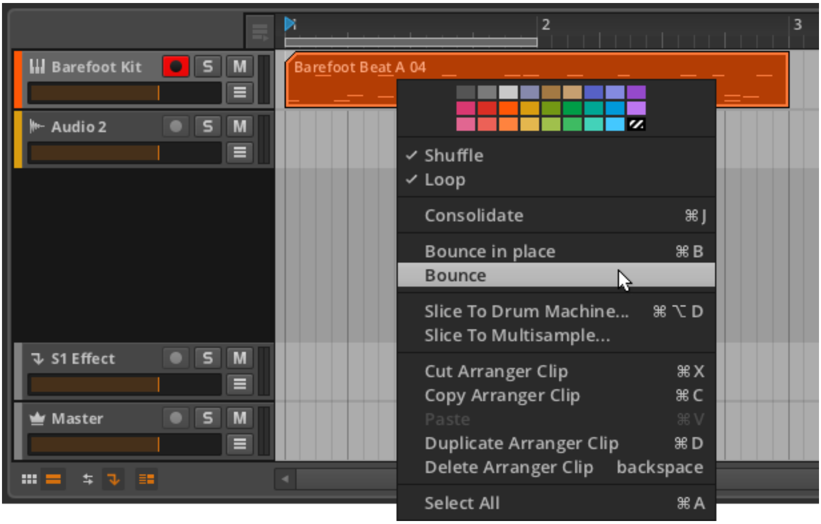
Note

General information about **Sampler** and every other Bitwig device can be found in chapter 15.

11.2. Bouncing to Audio

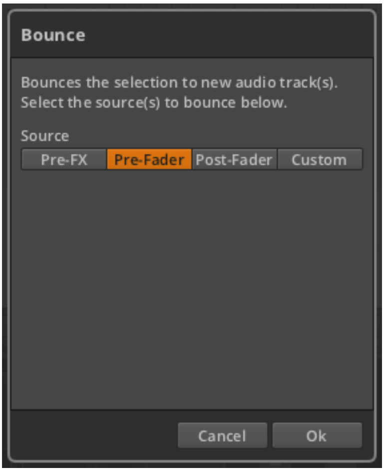
The concept of an audio *bounce* may be familiar. Sometimes called an *export* or *render* in other contexts, a bounce is a consolidated audio version of some part of your session. In this case, we want to investigate bouncing a note clip.

By right-clicking a note clip, a couple of bounce options are listed in the context menu. (These same options also appear in the *Edit* menu.)



11.2.1. The Bounce Function

The simple *Bounce* function presents a dialog box.



The choices refer to different places in the track's signal flow, and you get to select which point you would like the audio to come from.

The choices include:

› *Pre-FX*: The raw audio signal from the primary instrument's output.

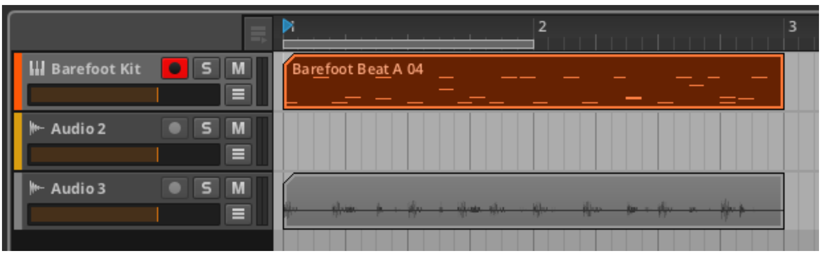
› *Pre-Fader*: The audio signal after the track's device chain but before the track's volume setting is applied.

› *Post-Fader*: The audio signal after the track's device chain and volume setting.

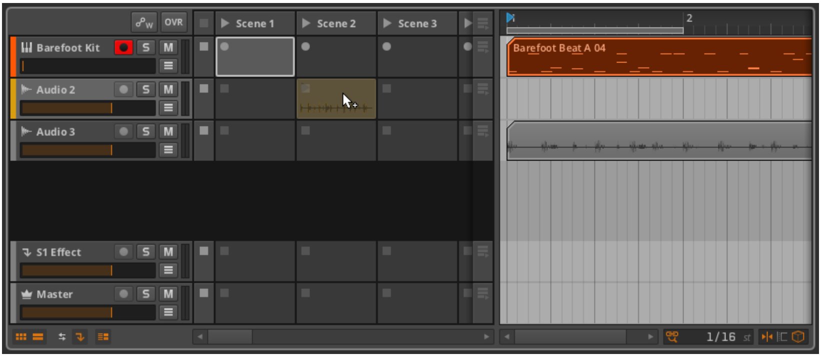
› *Custom*: A special menu of options that includes every top-level signal junction in the track, including from within the device chain.

In this example, the instrument track in question has three top-level devices: **Drum Machine**, **Delay-2**, and **Bit-8**. Selecting one of these options chooses the audio output of that device for the bounce.

After making your selection, click *Ok* to bounce the audio onto a new track.



If you want a standard, pre-fader bounce, you can also click and drag a clip while holding [ALT] ([SHIFT]+[CTRL] on Mac).



11.2.2. The Bounce in Place Function and Hybrid Tracks

The *Bounce in place* function is similar to the *Bounce* function with two key differences.

First, it presents no dialog box, taking the audio output from the primary instrument (*Pre-FX*).

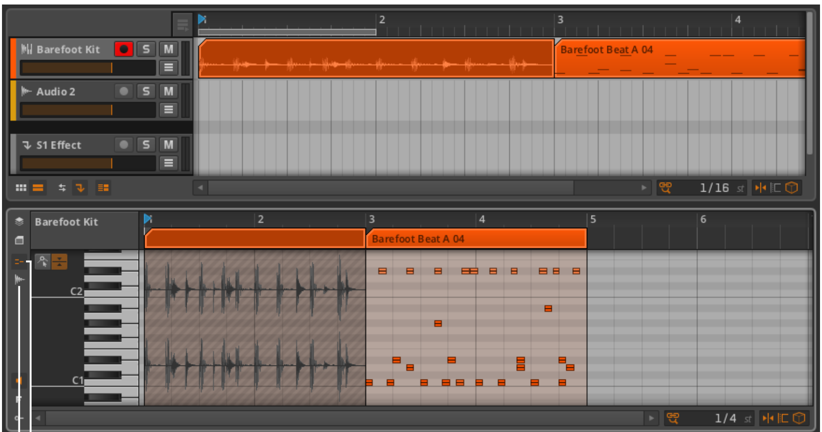
Second, it replaces the clip you are bouncing with the bounce itself.

Note

Since *Bounce in place* deletes your source clip, it is a good practice to copy the clip (perhaps to the Clip Launcher) before using this function.

Since this was the only note clip on the track, Bitwig Studio has converted it from an instrument track to an audio track while preserving the entire device chain.

If there were other note clips on the track, it would have been converted from an instrument track to a hybrid track.



Note Editor button

Audio Editor button

Since hybrid tracks allow both audio and note clips to be present, the **Detail Editor Panel** now has its *Audio Editor* and *Note Editor* buttons to keep things straight. These buttons (and the panel) work as they did when we first saw them in layered editing mode (see section 10.1.4). Otherwise, hybrid tracks work the same as instrument and audio tracks.

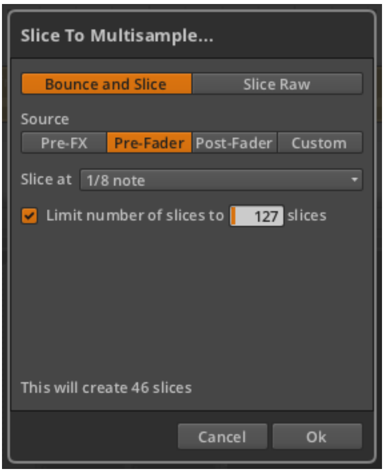
11.3. Slicing to Notes

The concept of a musical *slicing* operation may be familiar. The idea is to take an audio waveform and cut it into logical pieces that can be played with note messages.

By right-clicking an audio clip, a couple of slicing options are listing in the context menu. (These same options also appear in the *Edit* menu.)

11.3.1. The Slice to Multisample Function

The *Slice to Multisample...* function presents a dialog box.



The dialog begins with two options regarding the source to be sliced:

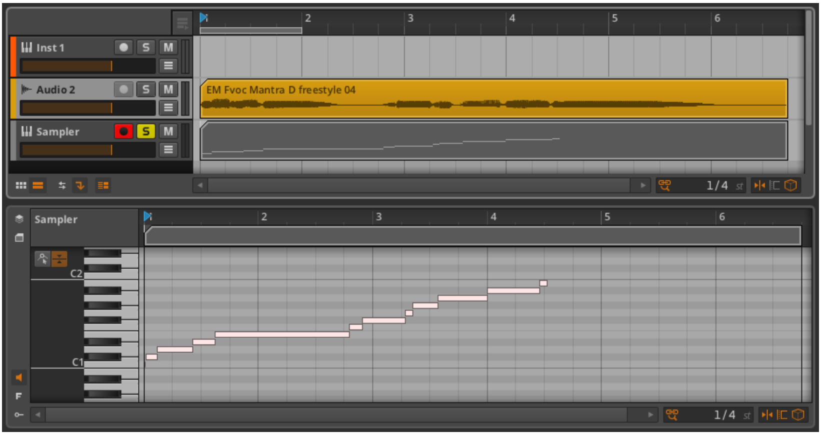
› *Bounce and Slice*: Executes a bounce function of the clip before slicing it. If this is selected, the signal flow options from the *Bounce* dialog are shown below (see section 11.2.1).

› *Slice Raw*: Simply slices the raw source event.

After these choices comes the critical *Slice at* setting, which determines at what interval slices will be made. The choices are self-explanatory, including event-based intervals (*Beat Marker*, *Onset*, and *Audio Event*) and time-based intervals (*Bar*, *1/2 note*, *1/4 note*, *1/8 note*, *1/16 note*, and *1/32 note*).

The final option, when enabled, allows you to limit the number of slices that will be made. This does not alter the *Slice at* setting, but simply stops slicing if the slice count set has been reached.

Choosing to *Slice Raw* at each *Onset* and clicking *Ok* would lead to a new instrument track with a new note clip.



On this new instrument track, a **Sampler** device has also been created with the corresponding slice of audio assigned to each note seen in the note clip.

The original audio clip could now be rearranged by editing the note events, or it could be reinterpreted on the fly by playing any of these notes in real time.

11.3.2. The Slice to Drum Machine Function

The *Slice to Drum Machine...* function leads to the exact same dialog as *Slice to Multisample...* and produces a new instrument track with a new note clip in the same way, but the instrument track is given a **Drum Machine** device with each slice loaded into its own separate **Sampler**.

The choice between **Sampler** and **Drum Machine** is really one of workflow. While **Sampler** places all slices in the same signal chain, the

Drum Machine gives you independent chains (and a unique **Sampler**) for each slice. If you want to process individual slices in different ways, you might favor the **Drum Machine**.

In the end — like so many things in Bitwig Studio — the choice is up to you and your personal preference.

12. Working with Projects and Exporting

The title of this chapter isn't meant to cause confusion. Yes, we have been working with projects for the majority of this document, but there are a few details about projects that we haven't covered yet, starting with some details about how Bitwig Studio manages project files.

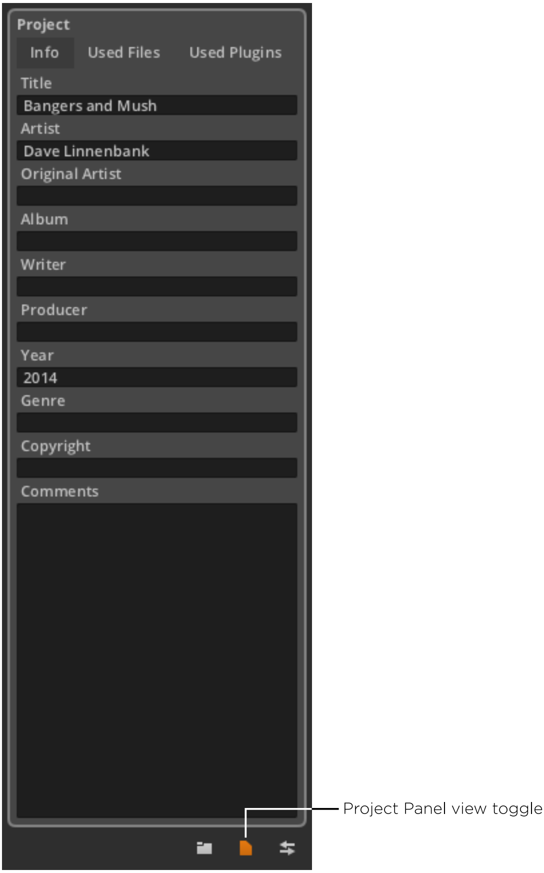
Each Bitwig Studio *project file* uses the *bwproject* extension. When you save a Bitwig Project file, the project file itself is placed in a new *project folder*. Whenever new content files are generated in a project, the program will automatically place them in the project folder within new sub-folders (such as *samples*, *plugin-states*, *recordings*, *bounce*, etc.).

While Bitwig Studio has its own preferences and settings, there are also project-based parameters that are stored within each project. And while preferences do apply across the entire program, these settings have to be reconciled with the content of your actual project file and folder.

In this chapter, we will look at the **Project Panel**, which manages your project's metadata and the status of files and plug-ins being used. We will talk a bit about the global groove settings and how they impact your project. We will show ways to share content between projects. And finally, we will examine exporting audio and MIDI from Bitwig Studio.

12.1. The Project Panel

The **Project Panel** is one of the "access panels" in Bitwig Studio.



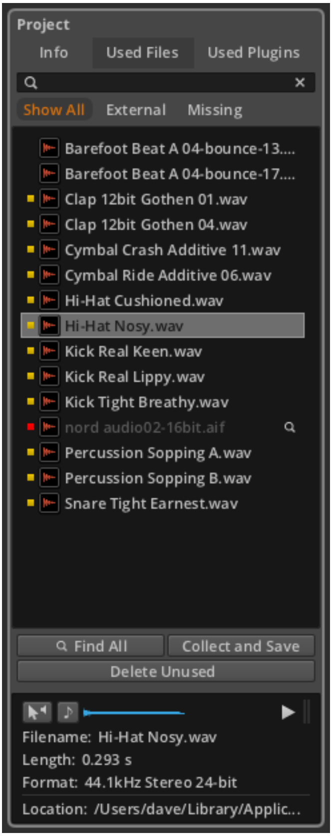
The purposes it serves are cleanly divided over three tabs.

12.1.1. Info Tab

The *Info* tab (shown above) allows you to fill in several metadata fields to describe your project. While you can use each of these fields as you see fit, their purpose is to help you store information and notes about each project file.

12.1.2. Used Files Tab

The *Used Files* tab lets you view and manage the audio files that are used by the current project.



The central focus of this tab is the list of audio files. At the top of the tab is a search field for narrowing the files being shown based on their name. And when one of the audio files is selected, an *info pane* will appear at bottom. This pane displays information about your file selection and offers a few options for auditioning files, similar to the **Browser Panel** (see section 4.1).

To the left of each audio file listed is either a yellow square, a red square, or a blank space. This indicates the file's status.

› A file with a blank space to its left is stored within the project's folder.

› A yellow square indicates that the file being used is *external*, or located outside of the project folder.

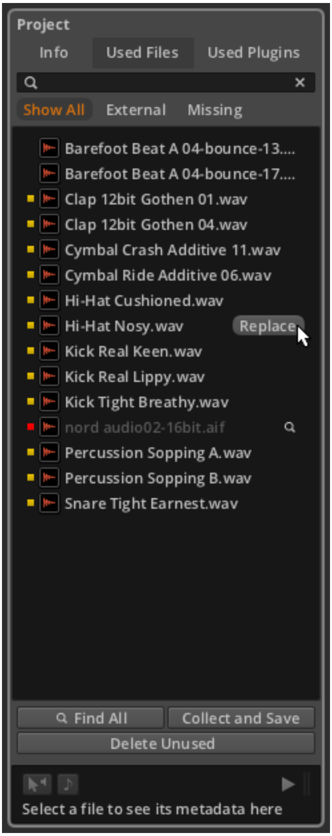
› A red square indicates that the file is currently *missing* and cannot been found. At the right edge of each missing file is a magnifying glass icon. Whenever a project has a missing file, its icon in the project tab section will include an exclamation point (*!*).

Files of all statuses will be shown when the *Show All* view button is enabled (it is the default). The other view buttons, *External* and *Missing*, show only files of those statuses when selected.

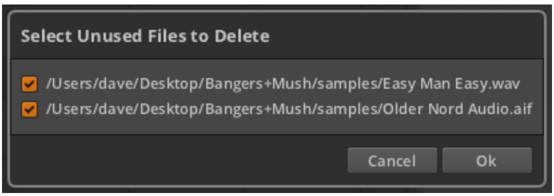
To search for a missing audio file: click the magnifying glass icon to the right of the file's listing. In the open file dialog that appears, navigate to the folder you would like searched, and then click *Open*.

To search for all missing audio files: click the *Find All* button at the bottom of the audio file list. In the open file dialog that appears, navigate to the folder you would like searched, and then click *Open*.

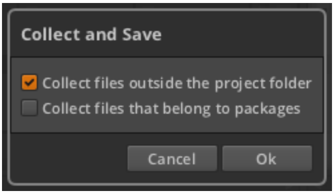
To replace one audio file with another: mouse over the file listing to be replaced, and click on the *Replace* button that appears on the right. In the open file dialog that appears, select the file you wish to replace it with, and then click *Open*.



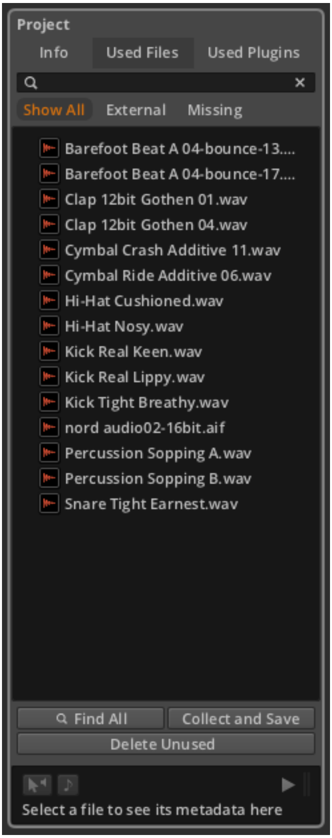
To delete unused files from the project folder: click the *Delete Unused* button at the bottom of the audio file list. In the dialog that appears, uncheck any files that you want to keep, and then click *Ok*.



To move external audio files into the project folder: click the *Collect and Save* button at the bottom of the audio file list. In the dialog that appears, select whether regular external files should be collected, and whether files within Bitwig Studio packages should be collected. Then click *Ok*.

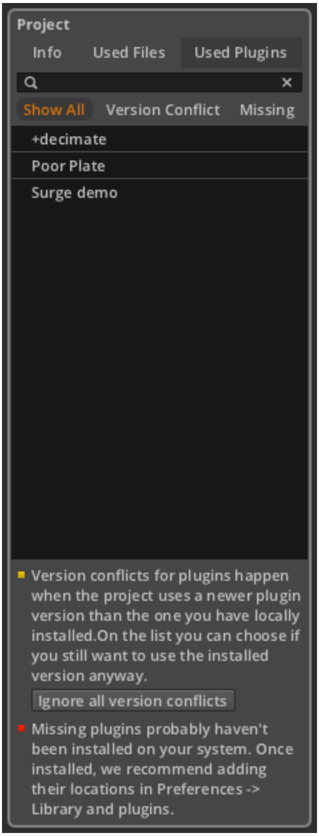


The *Collect and Save* function, found under *File* › *Collect and Save*. Depending on the options you have chosen, you can use this to quickly move all used audio files into the project folder.



12.1.3. Used Plugins Tab

The *Used Plugins* tab lets you view and manage the VST plug-ins that are used by the current project.



This tab is laid out very similarly to the *Used Files* tab. In this case, the central focus of the tab is the list of VST plug-ins. There is still a search field above the list. And to the left of each plug-in listed is either a yellow square, a red square, or a blank space.

› A plug-in with a blank space to its left is operating normally.

› A yellow square indicates that the plug-in has a *version conflict*. This means that the plug-in found on your system is an older version of the one that was saved in the project. When this happens, you can try to resolve it yourself, or you can ask Bitwig Studio to ignore the conflict.

To tell Bitwig Studio to ignore all plug-in version conflicts: click the *Ignore all version conflicts* button at the bottom of the plug-in list.

› A red square indicates that the plug-in used in your project is currently *missing* and cannot be found. When this happens, you can manually install the plug-in in question and make sure that the plug-in's location is known to Bitwig Studio (see section 4.1.7).

Plug-ins of all statuses will be shown when the *Show All* view button is enabled (it is the default). The other view buttons, *Version Conflict* and *Missing*, show only plug-ins of those statuses when selected.

12.2. The Global Groove

The musical idea of *shuffle* is to take a balanced (or "straight") rhythmic pattern and make every second note of the pattern a little late (or "swung"). The groove function in Bitwig Studio allows you to apply this idea so that notes which were programmed straight can be swung by a variable amount on playback. This function is nondestructive and can be adjusted or disabled at any time.

While each clip has local *Shuffle* and *Accent* settings (see section 4.2.5.4), the groove settings themselves are set at the project level.

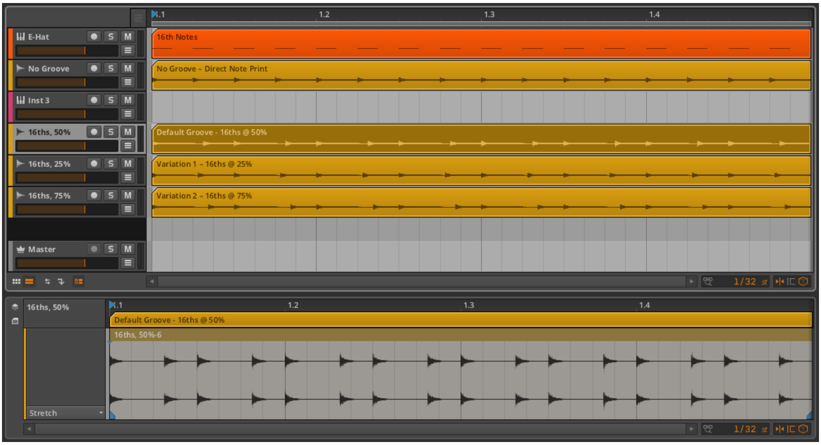
In the transport controls section of the window header are the *Global Groove* controls.

When the *Global Groove button* is enabled, the Global Groove settings will be applied to any clip requesting them. By clicking on the *Global Groove menu*, two groups of settings are available.

The *Shuffle* category has two settings:

› The *shuffle interval* determines whether groove will be applied at the *1/8* note or *1/16* note level.

› *Amount* sets the distance (from *0.00%* to *100%*) that even-numbered beats are delayed to the next lower beat division. So if the shuffle interval is set to *1/16* notes, the *Amount* setting determines how far each second 1/16 note is pushed toward the following 1/32 note.



In the above example, the source track is completely straight 1/16 notes (the *E-Hat* track). The three bottom audio tracks represent that source track printed with various amounts of *1/16* note groove applied.

The **Detail Editor Panel** focuses on the *50% Amount* example. Here, you can clearly see that each second 1/16 note is shifted halfway to the following 1/32 note.

The *Accent* category has three settings:

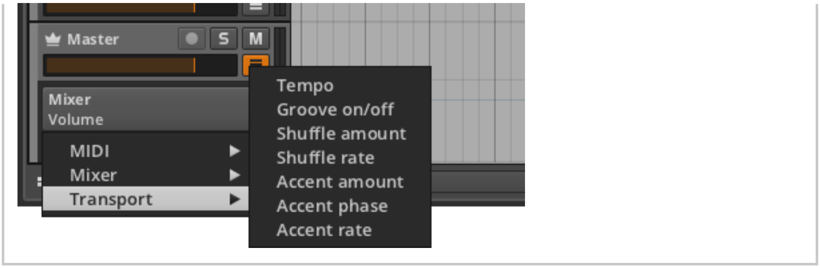
› The *accent interval* determines whether a slight emphasis is applied to every *1/4* note, *1/8* note, or *1/16* note.

› *Amount* sets the relative emphasis applied at the set interval. This is set between *0.00%* and *100%*.

› *Phase* sets an offset amount that the accent interval is shifted by. This is set between *-50.0%* and *50.0%*.

Note

All of these groove parameters can be automated in your project's master track under the *Transport* category. You can also automate the project's tempo from here.

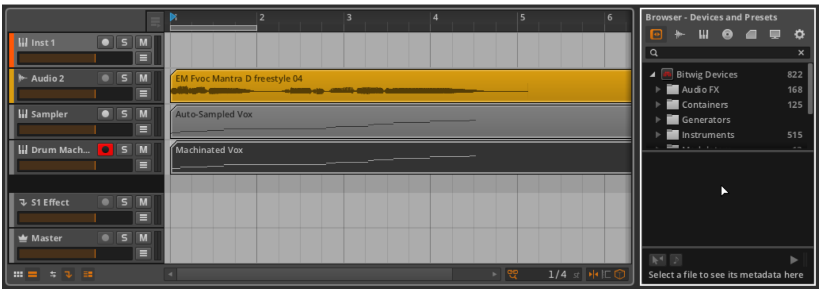


12.3. Working with Multiple Projects

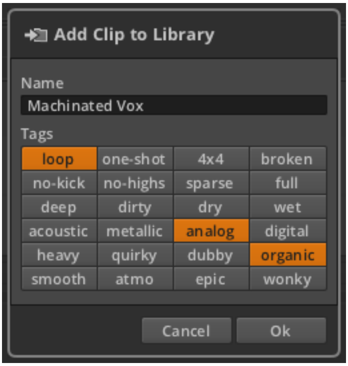
Bitwig Studio makes it quite easy to get your work from one project into another. This can be done either by storing your own library content via the **Browser Panel**, or by directly transferring data between open projects.

12.3.1. Adding Clips to the Browser Panel

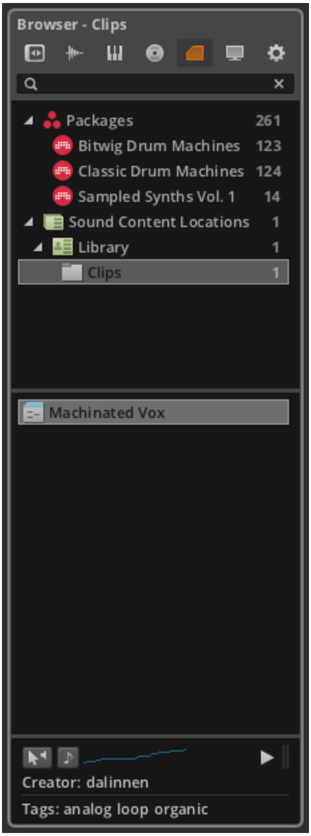
To add a clip to the library: click and drag the clip into the **Browser Panel**. In the dialog that appears, edit the clip *Name* if desired, enable any appropriate *Tags*, and then click *Ok*.



It doesn't matter which tab of the **Browser Panel** is showing when you drag your clip over. In fact, the **Browser Panel** doesn't even have to be called up before you begin dragging as you can call up any panel while using the mouse. In the case of the **Browser Panel**, you can press [B] any time to call it up.



Once you have stored your clip, it can be found and managed from the *Clips* tab of the **Browser Panel**.



Any clip stored in this fashion also contains its own parameters, the track's device chain, and any automation data.

12.3.2. Going Directly between Projects

Bitwig Studio allows you to have multiple projects open at the same time, with each open project represented in the project tab section of the window's header (see section 2.1.3.3). In addition to making it easy to quickly switch between projects, this also allows you to copy data between them.

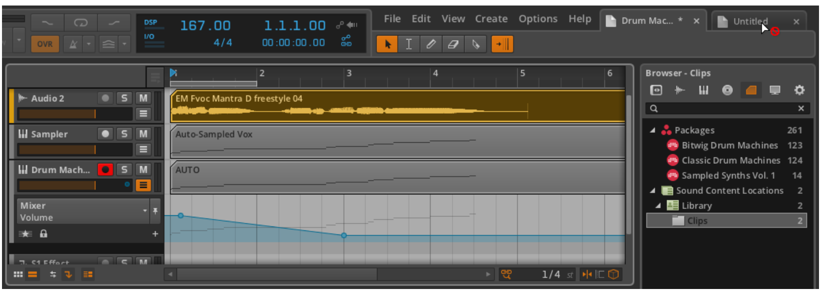
To transfer a clip(s) from one project to another: select and copy the clip(s) in the original session. Switch to the destination project, move the playhead to the desired insertion point (this can be done by clicking on either a Clip Launcher slot or at the position within the Arranger Timeline), and then paste.

> **Note**
>
> Copying and pasting clips within a session will maintain the original clip's automation but not its device chain. Copying and pasting clips between sessions will maintain neither.

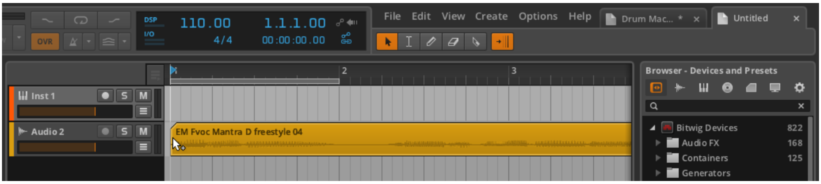
To transfer a device(s) from one project to another: select and copy the device(s) in the original session. Switch to the destination project, select the target track, and then paste.

The other option is to drag items directly from one open session to another.

To transfer an item(s) between two open projects: click and drag the item(s) from the original session to the target project's tab. While still holding the mouse, wait for the target session to load, and then drag and release the item in the appropriate location.



The cursor that includes a circle with a diagonal line through it indicates that releasing your item(s) on the project tab itself would do no good. Very quickly, the target project will load.

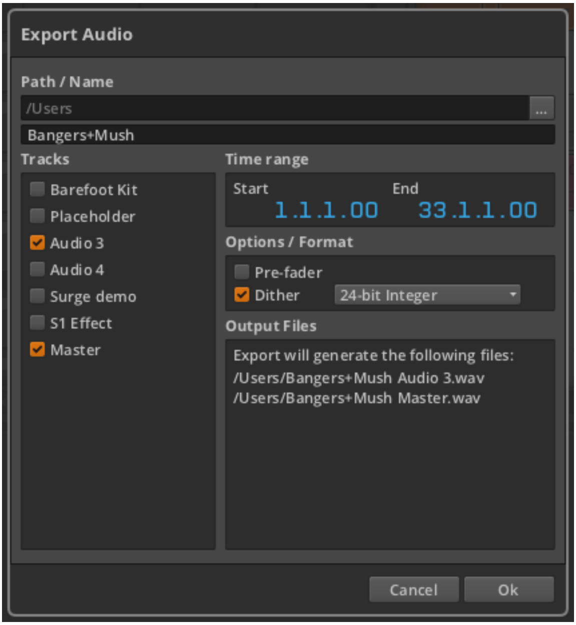


Note

Dragging clips between sessions will maintain only the clip, not the automation or device chain. Device(s) can be transferred separately using the same method.

12.4. Exporting Audio

To export audio from Bitwig Studio, either completed songs or individual tracks, use the *File* › *Export Audio...* function. When this option is selected, a dialog box with several sections appears.



› The *Path / Name* section assigns both the destination for the file(s) and the name stem to be used.

› To change the destination, click the ellipsis (...) button on the far right, and an open dialog will appear.

› To change the name stem, click in the lower text field and begin typing. By default, the project name is used for the stem.

› The *Tracks* section lists all tracks in the project. Check every track that you want to be exported. If you want to export the entire project (a completed song, for instance), check only the *Master* track.

› The *Time range* section determines the time range of the project to be exported. Both the *Start* and *End* parameters are set using song positions.

› The *Options / Format* section gives you three settings.

› To ignore all mixer volume automation, select the *Pre-fader* option. This can be especially helpful when exporting stems.

› Select *Dither* to add a very small amount of noise to your exports. This can help the lower-resolution export files best match the high-resolution internal signals of Bitwig Studio. This noise is generally inaudible.

› The *format menu* sets the format of audio to be exported. The choices include *16-bit Integer*, *24-bit Integer*, and *32-bit Floating-point*.

› The *Output Files* section lists the locations and names of audio files that will be created. When only one track is selected, the one file created will match the stem name chosen in the *Path / Name* section. When multiple files are being created, that stem will be followed by the source track's name.

Once the *Ok* button is pressed, the files will be created.

Note

As the *Time range* is set in song position value, only Arranger Timeline selections (not Launcher clips) can be exported in this way.

The *Export Audio* function dialog uses your current selection for its default settings. So if you want to export only a single clip from one track, first select that clip and then choose *File* › *Export Audio...*.

12.5. Exporting MIDI

To export MIDI from Bitwig Studio, choose *File* › *Export MIDI...*. In the save file dialog that appears, set the desired name and location for your MIDI file. This file will include all notes present in your project's Arranger Timeline, organized by track.

13. MIDI Controllers

MIDI controllers — or simply *controllers* — can be a critical part of any production environment or performance setup. Bitwig Studio supports MIDI controllers in general, whether you are playing in notes or you are mapping physical knobs and sliders to the program's parameters.

Bitwig Studio comes with various *controller scripts*. Each script is programmed for a specific MIDI controller, with a few scripts for *Generic* controllers of any make.

For the generic controllers, functionality is basic. If the controller has keys, you can send note messages. And if it has assignable knobs, you can map those knobs to any mappable control in Bitwig Studio or use them to manipulate the primary device's macro controls (see chapter 14.2.1).

For the controllers that are specifically supported, more functions are allowed. This can include control of track mixer functions, device macros and parameters, the transport, clip launching, and more. As each controller can vary greatly in size, shape, and functionality, the built-in mappings supported by Bitwig Studio also vary from controller to controller.

Note

Anyone with knowledge of JavaScript and the MIDI specification can customize any of the included controller scripts or even write their own. For full details on Bitwig Studio's controller API, select *Help › Control Surface Scripting Guide...*.

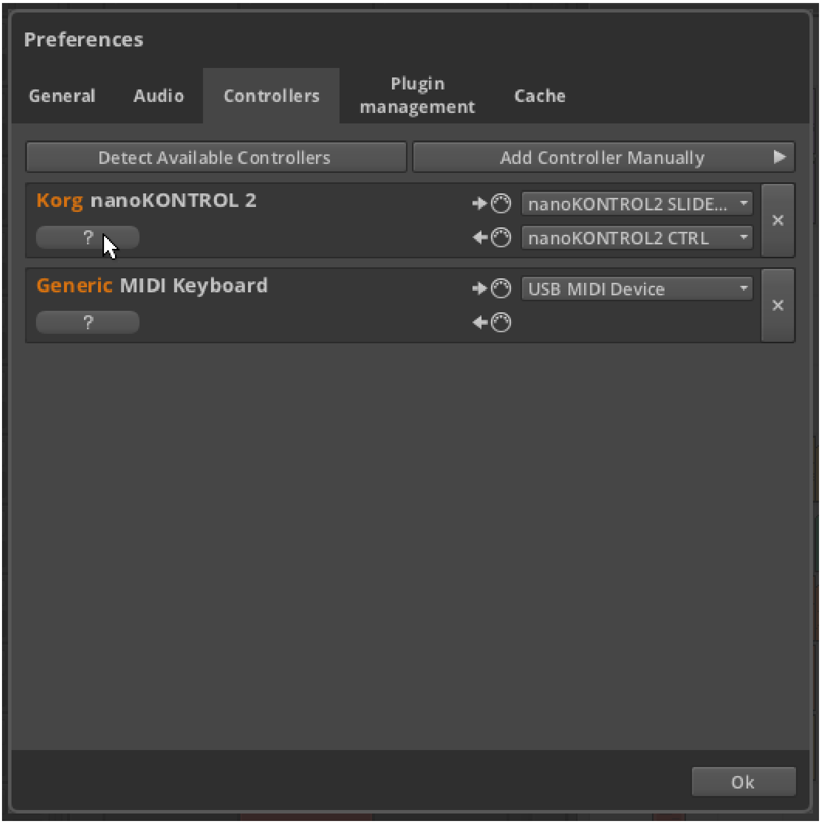
This chapter covers how to use both the default mappings for your controller (if supported), and how to manually assign and manage MIDI mappings.

13.1. Soft Controller Assignments

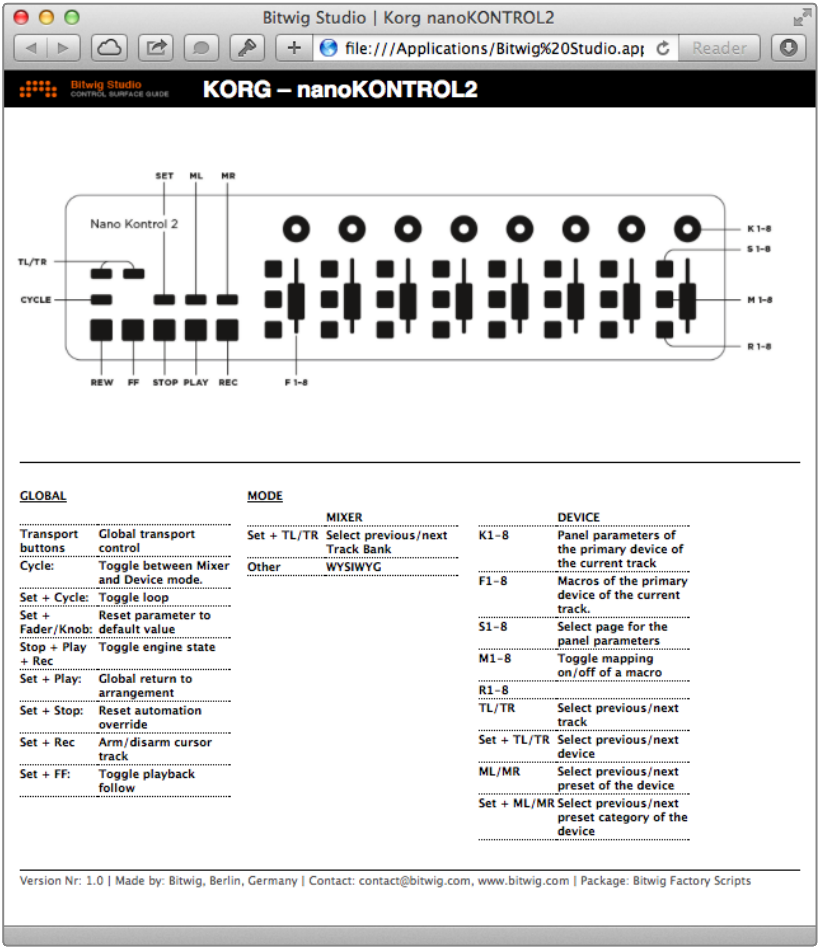
For any controller used with Bitwig Studio, certain default behaviors are available. We will start by seeing where documentation can be found for each controller. Then we will get to know the Device Mappings pane, which is available on every device.

13.1.1. Default Controller Documentation

Earlier, we registered our controllers on the first launch of Bitwig Studio (see section 0.1.2.2) and also referenced the Preferences window (see section 0.1.3) for using the same interface. By selecting *Options* › *Preferences*, or by pressing [CTRL]+[COMMA] ([CMD]+[COMMA] on Mac), we will call the Preference window's *Controllers* tab back up for another purpose.



For each controller type, Bitwig Studio has documentation that details how the hardware is set to interact with the program. This documentation can be accessed in your default web browser by clicking question mark (?) button beneath the listed controller.



Bitwig Studio | Korg nanoKONTROL2

file:///Applications/Bitwig%20Studio.app

Bitwig Studio CONTROL SURFACE GUIDE

KORG – nanoKONTROL2

GLOBAL

Transport buttons	Global transport control
Cycle:	Toggle between Mixer and Device mode.
Set + Cycle:	Toggle loop
Set + Fader/Knob:	Reset parameter to default value
Stop + Play + Rec	Toggle engine state
Set + Play:	Global return to arrangement
Set + Stop:	Reset automation override
Set + Rec	Arm/disarm cursor track
Set + FF:	Toggle playback follow

MODE

	MIXER
Set + TL/TR	Select previous/next Track Bank
Other	WYSIWYG

	DEVICE
K1-8	Panel parameters of the primary device of the current track
F1-8	Macros of the primary device of the current track.
S1-8	Select page for the panel parameters
M1-8	Toggle mapping on/off of a macro
R1-8	
TL/TR	Select previous/next track
Set + TL/TR	Select previous/next device
ML/MR	Select previous/next preset of the device
Set + ML/MR	Select previous/next preset category of the device

Version Nr: 1.0 | Made by: Bitwig, Berlin, Germany | Contact: contact@bitwig.com, www.bitwig.com | Package: Bitwig Factory Scripts

As mentioned above, the functions available are dependent on the type of controller being used.

13.1.2. The Device Mappings Pane

Note

The functionality described in this section will work only with non-*Generic* controllers.

As we discussed in chapter 7, all actual device control elements are found within the **Device Panel**. In this section, we will revisit the **Device Panel** to see how it facilitates soft controller assignments.

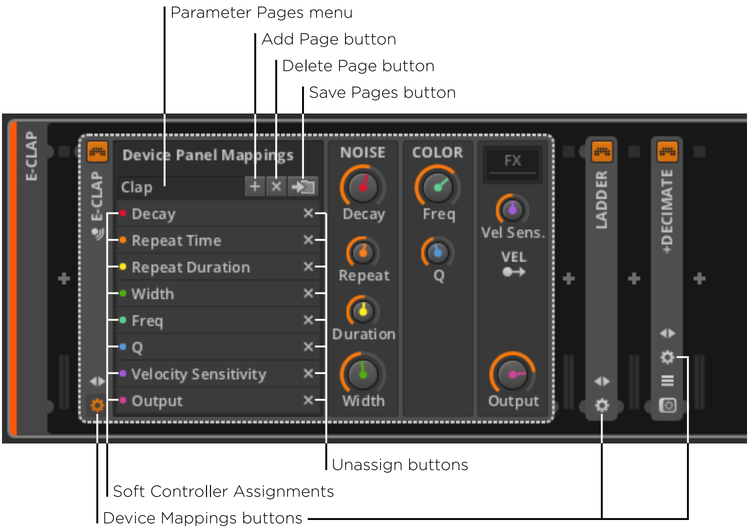
"Soft controller assignments" refers to controller assignments that can dynamically shift, following your focus on different tracks and devices within a project. For this functionality, only one device per track is designated as that track's *primary device*. So when that track is focused on in Bitwig Studio, its primary device will receive the incoming controller messages.

In the example above, the **Ladder** device has an icon beneath its title. This small circle with radiating arcs is the *primary device icon*, indicating that this is the primary device in this track.

To make a device the primary device: right-click on the device's vertical header and select *Set as Primary Device* from the context menu.

Once executed, the selected device becomes the primary device for its track. If you have a recognized MIDI controller connected and set up in Bitwig Studio, the device's interface may have some new color now.

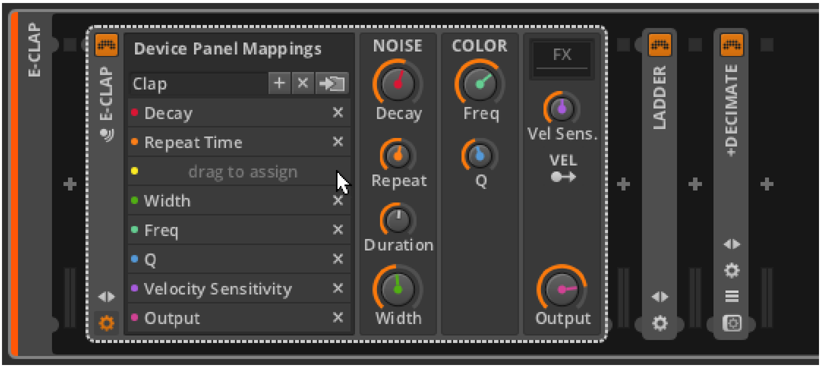
The colored interface items represent the eight current soft controller assignments. The details of these mappings are available in the device's Device Mappings pane, which is shown when the gear-shaped *Device Mappings button* is clicked.



The *Device Mappings pane* shows and gives access to the *soft controller assignments* that come with being the primary device for a track. Each assignment is represented here by a small color dot along with the full name of the parameter under control. Since these eight hardware controls will be used over and over again, they are always colored in rainbow order (red, orange, yellow, green, cyan, blue, indigo, and violet) to help you mentally connect each particular hardware control with its ever-changing assignment.

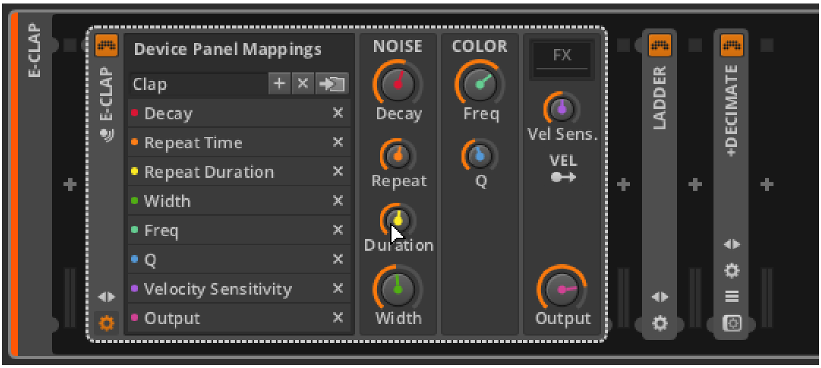
At the end of each assignment row is an unassign button, shown as an *x*.

To remove a soft controller assignment: click the assignment's unassign button.



In this example, the color overlay has been removed from the *Duration* knob, as has the parameter listing in the Device Mappings pane.

To make a soft controller assignment: click and drag the assignment's unassign button.

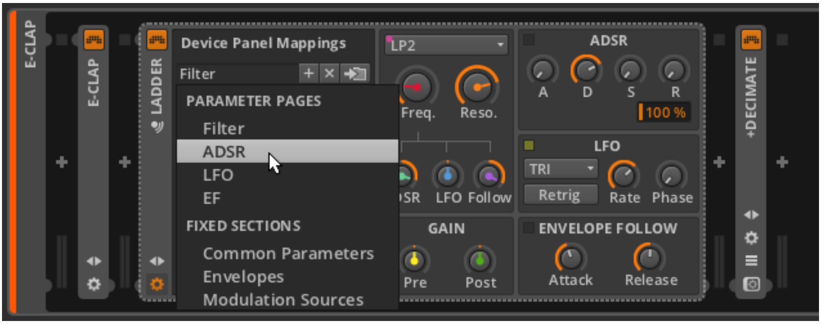


By dragging the blank, yellow assignment button onto the *Duration* knob, we have now connected the third soft controller to the *Repeat Duration* parameter. You could also replace an assignment this way, by dragging a currently assigned soft controller from the Device Mappings pane onto the new target parameter.

Since you may well want access to more than eight parameters of a device (or even an alternate set of soft controller assignments), each device contains multiple sets of soft assignments, called *parameter pages*.

At the top of the Device Mappings pane are four interface objects for parameter pages:

› The *Parameter Pages menu* displays the name of the current parameter page. By clicking the name, a menu loads allowing you to switch to another parameter page.



Devices with larger numbers of knobs tend to have a larger number of preset parameter pages.

› The *Add Page button* inserts a new, blank parameter page.

To rename a parameter page: [CTRL]-click ([ALT]-click on Mac) the parameter page's name.

› The *Delete Page button* removes the selected parameter page.

› The *Save Pages button* stores the current set of parameter pages with the device. For example, by clicking this button on a **Ladder** device, the current set of parameters pages will now be available on any **Ladder** that you load afterward.

Before we move on, let's consider a use of the rainbow order in another context. Most controllers that support soft controller assignments can also support a "mixer mode."

The following images demonstrate a project in **Mix View** both without and then with mixer mode engaged:

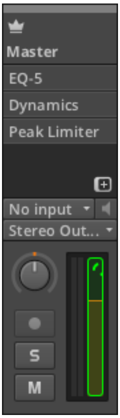
Notice how the volume and pan knobs for the first eight tracks are using the first eight soft controllers, as shown in rainbow order again.

13.2. Manual Controller Assignment

Any device with assignable hardware knobs/faders allows manual assignment of these controls to project parameters, such as device parameters or track mixer elements.

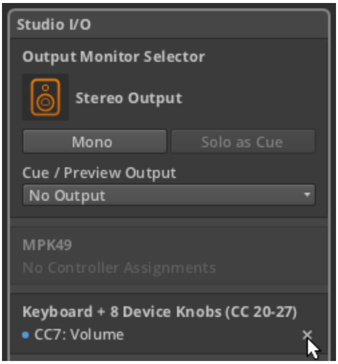
To make a manual controller assignment: right-click on the parameter you wish to assign and select *Learn Controller Assignment...* from the context menu. The targeted parameter will now be framed in bright green with an animated circle icon, indicating that you should "turn a knob." Then move the hardware control you wish to assign.

In this example, we've right-clicked on the master track's volume control.



After you move the hardware controller in question, the software parameter will return to its normal appearance, but the on-screen control will be moving as you move the physical control.

To remove a manual controller assignment: right-click on the parameter you wish to unassign, and select *Clear Controller Assignment* from the context menu. Or in the **Studio I/O Panel**, click the *x* button for the parameter assignment you wish to clear.



Finally, if you are using soft controller assignments, you can still make manual controller assignments. In this situation, any new assignments will override soft assignments that usually work in the current mode.

As an example, let's start from the "mixer mode" case from the end of the last section.

By manually assigning the master track's volume and pan controls to my hardware controller's eighth fader and knob, both of those master track controls will be colored violet, and the track that was previously using those controllers (*FX Storm*) will lose them.

In this example, my last fader and knob will always control the master track while my controller is in mixer mode.

14. Advanced Device Concepts

We have talked about and dealt with devices throughout this document. As we have seen, it's quite possible to operate devices in all the normal ways without delving into their advanced functionality. In this chapter, however, we'll explore device capabilities that are deeply powerful and generally unique to Bitwig Studio.

The aim of this chapter is not to educate you on any particular device or its parameters. While we will examine a few devices here in detail, our purpose is primarily to investigate concepts that are relevant to many devices. A separate reference section on the Bitwig devices themselves can be found at the end of this document (see chapter 15).

In this chapter, we will investigate nested device chains, we will examine Bitwig Studio's unique Unified Modulation System, and we will take note of some of the advanced VST plug-in options provided.

Congratulations; we've made it to the deep end of the pool. Now take a big breath.

14.1. Nested Device Chains

We discussed long ago how each track has its own device chain. Since then, there have been references to "top-level devices," meaning the devices that are directly in a track's device chain.

Most of the Bitwig devices actually possess one or more device chains of their own. These lower-level device chains, or *nested device chains*, solve several problems inherent to software-based music production.

For one thing, a single preset can contain vast configurations of devices, from a standard single device to something far more ornate. For another, the idea of nesting devices allows for unique signal routings that aren't usually possible in software, such as blending serial and parallel structures across a single device chain.

But we will return to device chains in a moment. Since the idea of parallel signal structures has already been mentioned, we should start this discussion with the humble, crucial *Mix* knob.

14.1.1. The Mix Parameter

For many audio effect processes, it is critical that the original, unprocessed sound is blended together with the affected sound. A good example is a simple delay effect. Hearing the original sound provides context for the delayed copy that follows. (A simple delay effect with no original sound mixed in could be better described as "late.")

To facilitate this blending, the idea of a *wet/dry* control is common in audio effects. This is usually implemented as a single knob that cross-fades between purely "dry," unprocessed signal at the minimum value, and purely "wet," post-processed signal at the maximum value, with every value in between representing a gradual blend of the two.

In Bitwig devices, this function is found on many devices via a parameter called *Mix*.

In the above example, we are using the **Freq Shifter** audio FX device, which is a frequency shifter. With the *Mix* parameter set to *33.3%*, a third of the device's output is the result of the frequency shifting process. This means that the signal received by the device (before any effect is applied) makes up the remaining two-thirds of the output, for a 2:1 blend of dry to wet signal. If *Mix* was set to *66.6%*, the balance would be reversed, with wet signal predominating at a 2:1 ratio.

So when you find a *Mix* parameter knob in the bottom right corner of a Bitwig device, it is providing this same wet/dry, parallel processing structure. In any of these cases, a *Mix* setting of *100%* would produce an output with no truly dry signal, and a setting of *0.00%* would effectively bypass the device by outputting only dry signal.

> **Note**
>
> If you find a *Mix* parameter knob that isn't in the bottom right corner of the device, it is carrying out a different function that is specific to that particular device.

Finally, *Mix* is not exclusive to audio FX devices and can be found on some devices in nearly every category. In the categories that don't use this *Mix* parameter (note FX and instruments), any incoming audio is generally passed directly to the audio outputs.

14.1.2. Container Devices

After starting with a simple in-line routing control, we will move on to nested device chains. And we will start with devices that are made to provide parallel device chains.

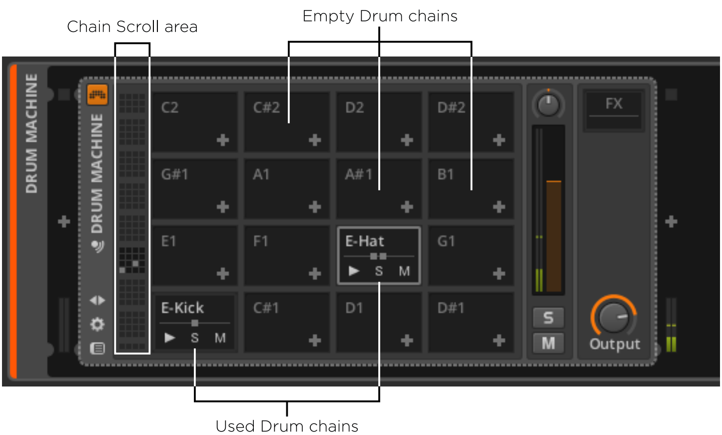
Container devices are utility devices whose primarily function is to host other devices. So while most devices contain some type of nested device chain, container devices couldn't exist without them.

Three particular container devices (**Drum Machine**, **Instrument Layer**, and **FX Layer**) came up in passing when we first saw the mixer's track fold button (see section 6.1.1), and the two "layer" devices reappeared indirectly when we discussed dragging devices to layer them (see section 7.4). Each of these devices allows for a large number of device chains within them.

We will examine each of these three container devices briefly.

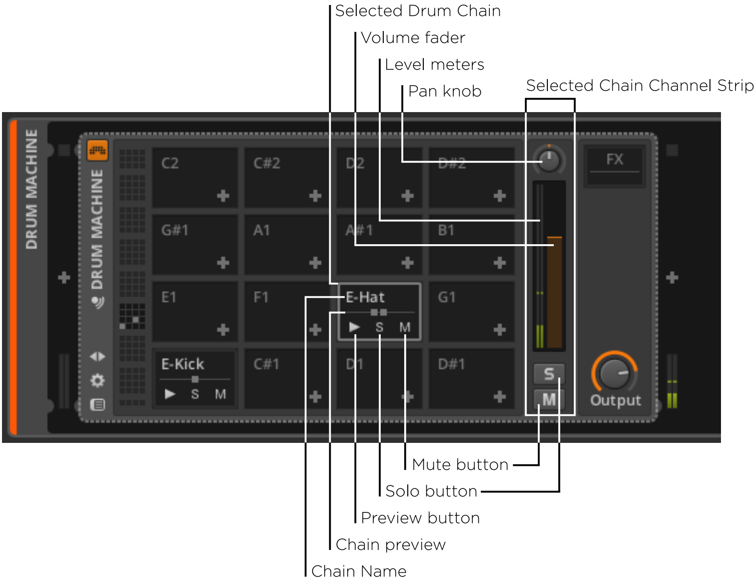
14.1.2.1. Drum Machine

Drum Machine is made to house multiple instruments, each of which will be triggered by a specific note message (for example, C1 for a kick drum, F#1 for closed hi-hat, etc.).



Corresponding with the 128 possible MIDI notes, **Drum Machine** offers up to 128 device chains, each called a *drum chain*. 16 drum chains are displayed at a time, and the *chain scroll area* on the left allows you to click or scroll the focus to a different set of chains.

An empty drum chain simply displays the note that it responds to and an *Add Device button* (+) for loading a device directly into that chain.



Used drum chains each have their *chain name* listed at top, and at bottom are a *preview button*, a *solo button*, and a *mute button*.

To the right of the displayed drum chains is the *selected chain channel strip*. Whichever drum chain is selected is surrounded by a blue-green border, and this area of the device provides a small channel strip for that chain, including larger solo and mute buttons, a *volume fader*, a *pan knob*, and *level meters*.

Every used drum chain also has a small *chain preview* displayed across its middle. This central line with squares placed along it is a silhouette of the drum chain, with the squares representing the number of devices currently at the top-level of the drum chain.

Note

A maximum of seven squares fit within this small chain preview area, but more devices may be added to the drum chain.

To view an individual chain: either double-click the chain, or select the desired chain and enable the *Chain View button*.

Chain View button

What can now be seen is the drum chain itself, which is, again, a device chain. The two squares from the chain preview were representing these **E-Hat** and **Delay-1** devices, which have the exact same interfaces we are accustomed to.

To accompany the blue-green border of the selected drum chain on the left, the full drum chain here is also tinted a slightly duller version of the same color, illustrating the nested nature of that specific chain and its devices.

To reiterate this idea, the **Delay-1** device is currently within this drum chain. This means that only this particular instrument (triggered by F#1) will have this device applied to it.

If I were to move this device to the right and out of the drum chain, it will now be in the track's device chain just after the **Drum Machine**.

Accordingly, all audio coming out of **Drum Machine** is now being affected by **Delay-1**.

14.1.2.2. Instrument Layer

Instrument Layer is made to house multiple instruments, all of which will be triggered by any incoming note message. The general effect of this device is to make layered sounds or "stacks."

The chains in this device can be called *instrument chains*. Each is still representing a full device chain, but unlike **Drum Machine**, there is no set number of chains. Because of this, there is only one *Add Device button* in the main interface of **Instrument Layer**, with each added device being placed on a newly created instrument chain. If enough instrument chains are added, the chain list itself can be scrolled vertically.

Each instrument chain has its own built-in channel strip, quite similar to each track header in the **Arranger Timeline Panel**. Also as in the Arranger, the selected chain is given a silvery tint.

14.1.2.3. FX Layer

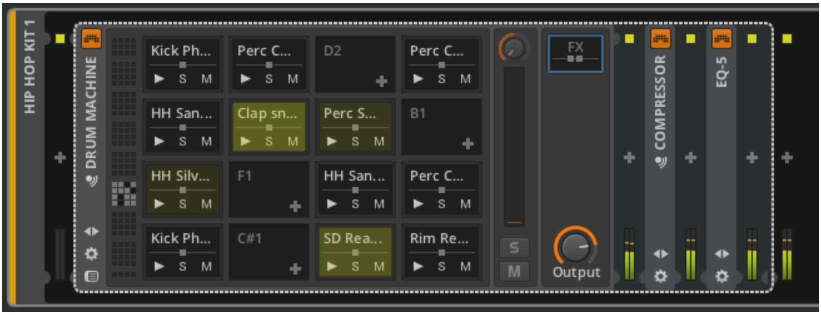
FX Layer is virtually identical to **Instrument Layer** except it is made to house a layer of FX chains.

14.1.3. Other Common Device Chain Types

There are several other types of nested device chains within Bitwig Studio. Some appear rarely or only once, but a few are reused multiple times.

Some of the most common types of nested device chains include:

› *FX* (or *Post FX*): A nested device chain for processing the device's entire audio output. The only difference between placing effects in this device chain instead of after the device is that this chain is fully stored with this device, which makes moving the device along with its modifiers (or saving presets) much easier. This chain type is mostly possessed by instruments and containers for instruments.



Post FX chains work in exactly the same way, but tend to show up on devices where other chains occurred first.

› *Pre FX*: A nested device chain for processing signal immediately before it enters the device.

› *Wet FX*: A nested device chain that processes only the wet portion of the device's output. The dry signal skips this chain and is mixed back in afterward. All devices with this chain also have *Mix* parameter knobs.

› *FB FX*: A nested device chain that is placed within the device's feedback loop. This is common on delay devices.

Note

Just like Bitwig devices, VST plug-ins can be used in any device chain at any level.

14.2. The Unified Modulation System

In sound synthesis, *modulation* is the idea that one component can influence another in a controlled way. For a simple musical example, think of vibrato (the subtle bending of pitch back and forth). To achieve this with synthesis, we often connect the output of a low-frequency oscillator (LFO) to a pitch input of an oscillator. The frequency of the LFO determines the rate of the vibrato, and the level of the LFO's signal determines the depth of the modulation.

Modulation can lead to elements that automatically change over time, based on assigned parameters and preexisting control sources. Some would say that modulation leads to more interesting and efficient results in sound programming. These are both good points.

In the days of modular hardware synthesizers, each modulation was highly visible as it was achieved by a patch cord properly connecting two modules. But in our era of computer-based music production, we see knobs on screen far more often than patch cords, and assigning (or even showing) modulations has become a real challenge. Many different interface models have been attempted, but no standard has been found.

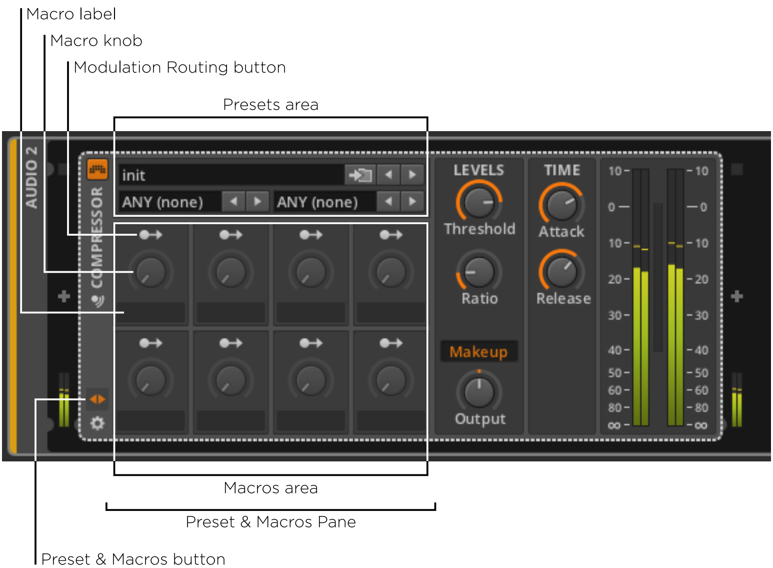
Bitwig Studio has its own unique, program-wide method for dealing with modulations. This *Unified Modulation System* allows you to easily assign and edit modulations (so you don't get stuck with fixed modulation routings). It also preserves parameter control as often as possible (so the modulated parameter's knob can still be used, allowing you to easily shift the modulation range). Even the current value of a modulated parameter is visible with this Unified Modulation System.

In this section, we will first explore the Unified Modulation System in device macros. We will then see the same principles used to assign modulations within an instrument. Finally, we will look at the modulator device category, where we will find independent control sources for modulating other devices.

14.2.1. Device Macros

Conceptually, a *macro* is a single function that yields multiple results. In the field of sound synthesis, this tends to appear as a single control that is mapped to various parameters of a device, a function that is available in Bitwig Studio.

While we previously looked at the *Presets & Macros pane*, at that time we only examined the presets area (see section 7.2.2). By clicking the *Presets & Macros button* of any device, the Presets & Macros pane will appear.



In the *macro area* are the eight macros allocated to this device. Every device has its own set of macros, and each of the eight macros on a device is identical. They are numbered from left to right by row, so the top row goes from *Macro 1* to *Macro 4*, and the bottom row contains *Macro 5* through *Macro 8*.

Within each macro are the following interface items:

› The *modulation routing button* (which resembles an output port with a patch cord coming out of it, awaiting connection) toggles to a mode where you can select destinations and set modulation amounts for this modulation source. When enabled, the button itself begins flashing, all currently assigned destinations become brightly colored, and all potential destinations are shaded.

› The *macro knob* sets the current value for the macro. And like nearly all device parameters, each macro knob can be MIDI mapped, automated, or (in some cases) modulated.

› The *macro label* is an editable description of the macro. It is here for your use.

The purpose of each macro is to modulate one or more parameters, with each parameter modulated across a specific range. This can give you live performance controls for a parameter's "sweet spot," create single knobs that create various changes across many parameters, give you a simplified interface for a device, or whatever else you decide might be useful.

To map a macro knob to a parameter: enable the macro's modulation routing button. Then click the target parameter and drag its value to set the point of maximum modulation.

The *Ratio* parameter now has a defined range for its modulation by *Macro 2*. The parameter's knob indicate the range visually, and the pop-up beside the cursor tells us that the maximum modulation is set to *+0.33*, or up an additional 0.33 points in the ratio. Thus the range is set relative to the parameter's current value and in the parameter's units.

Note

Because the modulation range is set relatively, the range displayed on the knob is also relative and does not directly correspond to the parameter's values. So you can twist the modulation range past the parameter's normal range, and this is correct.

You can assign additional parameters in the same fashion.

So in this **Compressor** example, turning up *Macro 2* will increase the *Ratio*, decrease the *Threshold*, and increase the *Output*. Essentially, we have a one-knob compressor.

You will notice that the label for this macro is *Ratio*. This label was created by Bitwig Studio when *Ratio* was the first parameter assigned to this macro.

To change a macro's label: double-click on the label and edit the text.

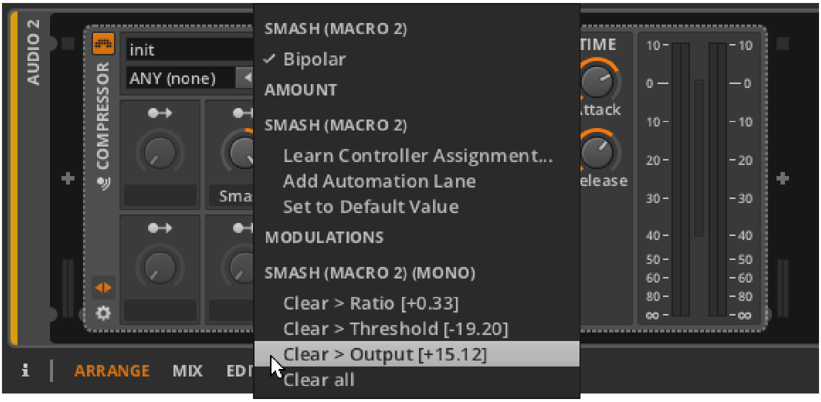
Since macros have no fixed assignments, they always operate in units of percentage. After disabling the modulation routing button, we can now snapshot the device with *Macro 2* set to *0.00%*,*50.0%*, and *100%*, respectively.

As you can see, the parameter knobs for *Threshold*, *Ratio*, and *Output* haven't moved at all, but their orange *parameter value indicators* have. While these indicators look like extensions of the knobs when no modulation is occurring, they are actually informing you of the parameter's current value after all modulation signals are summed and applied.

To switch a macro to bipolar mode: right-click the macro's modulation routing button, and then select *Bipolar* from the context menu.

A bipolar macro makes adjustments around a center value of *0.00%*, with *100%* at the far right and *-100%* at the bottom of the range. If you only set the macro to positive values, everything behaves exactly as it did before. The difference is that you can now move the device negatively, allowing you to move each modulated parameter by the maximum amount but in the opposite direction.

To clear a macro knob's parameter mapping: right-click the macro's modulation routing button, and then select *Clear* from the desired parameter.



This can also be accomplished by right-clicking on the parameter.

To clear all of a macro knob's parameter mappings: right-click the macro's modulation routing button, and then select *Clear all* from the desired parameter.

14.2.2. Modulations within a Device

Several devices have their own built-in modulation sources. Instrument devices are the best examples so let's look at Bitwig Studio's synthesis flagship, **Polysynth**.

On the right half of **Polysynth** are two columns of modulation routing buttons, each of which represents a modulation source that can be used to affect parameters within the synthesizer. The 10 modulation sources on this device can be broken down into three basic kinds:

› *Note expressions*: *VEL* (velocity), *KEY* (micro-pitch), *TMB* (timbre)

› *MIDI messages*: *MW* (modulation wheel), *AT* (channel aftertouch)

› *On-board control sources*: *LFO1* & *LFO2* (two monophonic LFOs), *PLFO* (a polyphonic LFO), *FEG* (filter envelope generator), *AEG* (amplifier envelope generator)

The note expressions should already be familiar (see section 10.1.2), and the MIDI messages are often transmitted by controller keyboards.

The on-board control sources are control modules in **Polysynth**'s architecture and can be freely routed. The two envelope generators each have a single hardwired modulation target — *FEG* to the filter's cutoff frequency, *AEG* to the volume amplifier — but they can be used for other modulations as well. Each of the LFOs has no hardwired target so their usage is completely left to you.

In the context of modulation sources, a monophonic source generates only one control signal that is then applied to all targets identically (musically speaking, *unison*). Monophonic sources are tinted a bright blue when selected, and they can even modulate nested devices, such as the **Filter** device shown below.

Polyphonic sources produce multiple control signals, potentially providing a unique signal for each note event (musically speaking, *divisi*). This is the same idea we experienced with expressions before, where each note contained its own, concurrent curve. Polyphonic sources are tinted a light green when selected, and they cannot modulate nested devices.

If you look back at device macros, you will notice that they are always blue, or monophonic. This makes sense since there is only one copy of each macro knob that applies equally to each note (unlike micro-pitch curves, for example).

Other than these few points, everything we saw about working with device macros applies here.

14.2.3. Modulator Devices

Finally, all devices in the modulator category are made to manipulate the parameters of other devices found within the modulator's *FX* device chain. In essence, modulators are free-standing control sources.

To borrow the example from the top of this section, we could use **LFO MOD** to give the usually static **Test Tone** a broad vibrato.

To see the settings and issues involved, we will go through another example from scratch.

Let's use the **Audio MOD** device here. It will allow us to approximate the envelope of the incoming audio signal and then apply that as a modulation source.

A modulator has no effect on the signal chain it is in; it simply passes any note and audio signals it receives directly to its outputs. For example, with the **Audio MOD** device, its filter component (*PRE-FILTER*) is only used internally to generate the control signal. The device itself is outputting a fully dry signal.

We can see that the **Audio MOD** device has a modulation routing button for the source *MOD*. This is where the control signal will come from so we can enable this button to make routing assignments.

Well, that is interesting. *MOD*'s modulation routing button is flashing blue (a monophonic source), but no parameters are available as targets. But this is correct because modulations can only reach nested devices, and every modulator device has an *FX* chain precisely for this purpose. By placing devices into a modulator's *FX* chain, they become potential modulation targets.

To move an existing device into a different device chain: click the device's header, and then drag the device into the chain.

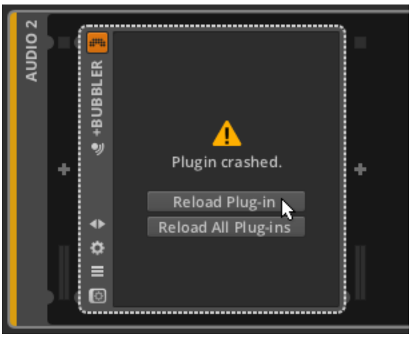
Once they are moved, the parameters of these devices can now be modulated.

Nesting devices within a modulator does not alter their behavior in any other way.

14.3. VST Plug-in Handling and Options

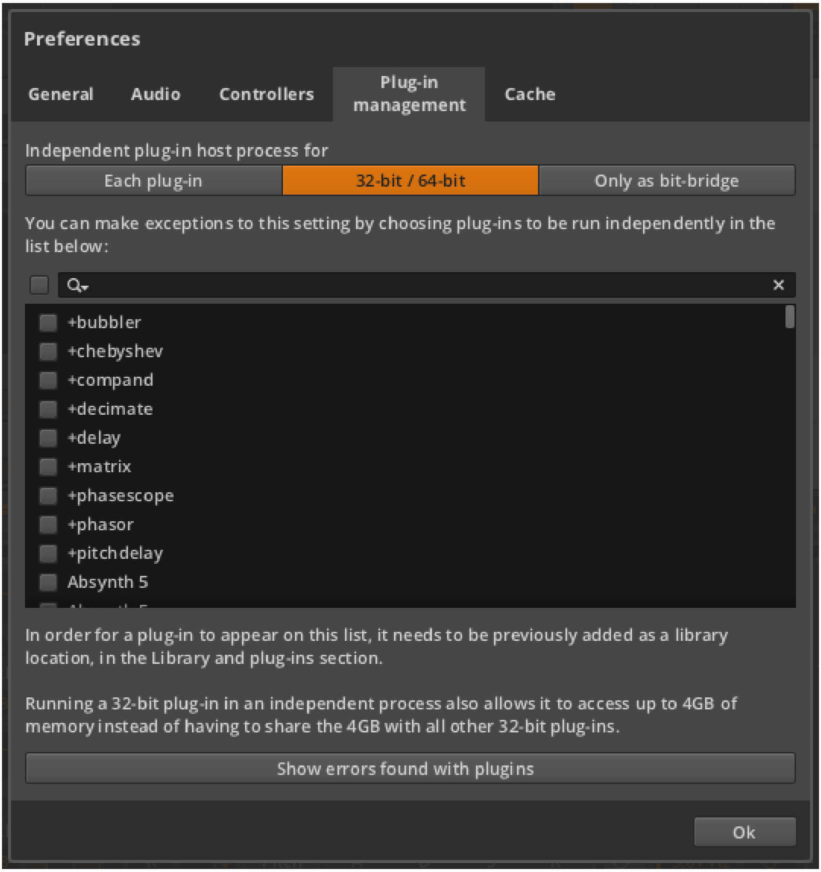
Speaking technically, Bitwig Studio handles VST plug-ins in a separate process from the program itself and its audio engine. This greatly reduces the chance of a plug-in crash also crashing the program itself. In many cases, a plug-in crash will happen discreetly, even allowing other track to continue playback.

If a plug-in does crash, its interface in the **Device Panel** will be replaced with a notification.



By clicking *Reload Plug-in*, the plug-in will be freshly called up again. Clicking *Reload All Plug-ins* will reload every crashed plug-in and leave those that haven't crashed alone.

Additionally, Bitwig Studio gives you the option to have all plug-ins handled together or to have each plug-in instance operate in its own separate process. In the Preferences window is a tab called *Plugin management*.



The three long buttons at the top of this tab are a toggle group that determines how plug-ins are handled. In the picture above, the *32-bit / 64-bit* setting is currently enabled so all 32-bit plug-ins are run in one grouped process, and all 64-bit plug-ins are run in their own grouped process. If one plug-in crashes, the group it is in also crashes, but the audio engine will continue running. This option is a middle ground in terms of both CPU resources required and stability provided.

If the *Only as bit-bridge* option were selected for a 64-bit operating system, then 64-bit plug-ins are loaded within the audio engine, and any 32-bit plug-ins are loaded within their own single process. (Within a 32-bit operating system, only 32-bit plug-ins are available, and they would all be loaded within the audio engine.) This option requires the fewest CPU resources.

If the *Each plug-in* option were selected, each plug-in is loaded within its own process, isolating every plug-in and the audio engine so that any single crash won't affect the other ongoing processes. This option requires the maximum CPU resources while also providing maximum stability.

The list of plug-ins below allows you to select individual plug-ins that should be given their own process for operating, effectively overriding global setting above. This could be useful if you want one or two plug-ins to have access to additional memory, for example. The search box just above the list allows you to quickly find plug-ins from the list. If the global *Each plug-in* setting is selected above, then the selections in this plug-in list are ignored.

15. Device Descriptions

This appendix provides a short description of each device that comes with Bitwig Studio. The devices are organized by type. Information on using devices can be found in chapter 7, and chapter 14 provides an explanation of more advanced device concepts. For specifics on any particular device's parameters, note that full parameter names are given as you click and adjust the parameter, and some parameters present tooltips when you hover your mouse over them.

15.1. Note FX

Each *note FX* device manipulates the incoming note signals before passing them onward.

15.1.1. Arpeggiator

An arpeggiator, which rhythmically cycles through the notes being held in a set order. For each beat, the specified note(s) is output with a set velocity and for a set duration.

15.1.2. Diatonic Transposer

A note transposer, which can correct or remove notes that do not match a set key and mode. Notes can also be shifted before the transposition is applied.

15.1.3. Note Filter

A filter that allows only notes from a set range of pitches and velocities (inclusive) to pass.

15.1.4. Note Pitch Shifter

A simple note transposer, which can shift the incoming notes by a set number of octaves and/or semitones. A *Fine* control is also provided for shifting by fractions of a semitone.

15.1.5. Transposition Map

A note transposer, which can remap each note class (for example, so every D becomes an F#, etc.). Notes can also be shifted before the transposition is applied.

15.2. Instruments

Each *instrument* device uses incoming note messages to synthesize audio output.

15.2.1. E-Clap

A monophonic electronic clap instrument made from noise, a low-pass filter, and repetitions.

The *NOISE* section comprises the instrument's sound generation parameters. The amplitude for the instrument is controlled by an AD envelope that has a short, fixed attack time and an exponential, adjustable *Decay* time.

Each incoming note message immediately triggers the amplitude envelope. And for the *Duration* time following the beginning of each note, the envelope is retriggered at every *Repeat* time interval.

For example, if *Duration* is set to *45ms* and Repeat is set to *10ms*, each note will trigger the amplitude envelope five times: zero milliseconds (the instant the note is received), 10ms, 20ms, 30ms, and 40ms.

Width sets the amount of stereo flutter added to each noise burst.

The *COLOR* section provides controls for the instrument's low-pass filter. *Freq* sets the cutoff frequency, and *Q* sets the amount of resonance.

The final section offers a control for the instrument's *Vel Sens.*(itivity) and a level control for its *Output*.

Modulation Sources:

› *VEL*(ocity) [monophonic] - The velocity portion of incoming note messages.

Nested Device Chains:

› *FX* - A chain for processing the device's entire audio output.

15.2.2. E-Hat

An electronic hi-hat instrument made from a blend of FM synthesis, noise with a comb filter, and a one-band equalizer.

The section at the top left contains *Attack* and *Decay* times for the AD envelope, along with a contour control for the shape of the decay segment. This global envelope shapes the output of the entire instrument.

The *COMB* section governs the comb filter that processes the noise generator's output. Parameters include cutoff *Freq*(uency), a bipolar *Feedback* control, and the wet/dry *Amount*.

The *FM HIT* section provides a classic FM operator pair for creating the impact sound of the hi-hat. The *Freq* knob at left sets the carrier's frequency, while the numeric controls beneath the X–Y grid show and set the modulator's frequency and amount, respectively. (These parameters of the modulator can also be set via the X–Y grid.)

The *FM HIT* section also has its own AD envelope, which has a short, fixed attack time and an exponential, adjustable *Decay* time. (Note that a longer decay setting may be interrupted if the global AD envelope has a shorter overall duration.) Finally, the *Mix* knob controls the balance between the noise and FM portions of the instrument.

The *EQ* section controls a simple high-pass filter. The cutoff frequency is set by the *Lowcut* knob, and the numeric control represents the filter's Q.

The final section offers a control for the instrument's *Vel Sens.*(itivity) and a level control for its *Output*, along with a *Width* setting for the amount of stereo flutter added to each noise burst.

Modulation Sources:

› *VEL*(ocity) [monophonic] - The velocity portion of incoming note messages.

Nested Device Chains:

› *FX* - A chain for processing the device's entire audio output.

15.2.3. E-Kick

An electronic kick drum instrument with optional pitch modulation.

The *GEN* section contains parameters for controlling and processing the instrument's slightly rectified sine oscillator. The frequency of this oscillator is set by the *Tune* knob, and its level is controlled by an AD envelope that has a short, fixed attack time and an exponential, adjustable *Decay* time. The *Click* option adds impact to the sound by doubling portions of it, and the *Tone* control sets the cutoff frequency of a gentle low-pass filter.

The *P. MOD* section concerns a separate AD envelope generator that controls the oscillator's pitch. You can adjust the *Amount* of modulation in semitones, the *Decay* time, and the shape of that decay segment with the contour control.

The final section offers a control for the instrument's *Vel Sens.*(itivity) and a level control for its *Output*.

Modulation Sources:

› *VEL*(ocity) [monophonic] - The velocity portion of incoming note messages.

Nested Device Chains:

› *FX* - A chain for processing the device's entire audio output.

15.2.4. E-Snare

An electronic snare drum instrument made from two tunable oscillators, a noise generator, and resonant high- and low-pass filters.

The *OSC 1* section houses the primary sine oscillator, whose frequency and decay time can be set directly with the *Tuning* and *Decay* knobs, respectively.

The *OSC 2* section contains a secondary sine oscillator whose settings are relative to oscillator 1. Accordingly, the frequency of oscillator 2 is set as an *Offset* from oscillator 1 in semitones, and oscillator 2's decay time is set with the *Decay X* parameter as a percentage of oscillator 1's decay time.

The *NOISE* section contains parameters related to the noise generator. This includes *Attack* and *Decay* times for the AD envelope that controls level, along with a contour control for the shape of the decay segment. And the *Width* knob sets the amount of stereo flutter added to each noise burst.

The *MIX* section is for controlling the balance between the three generator elements. *Osc* controls the balance between oscillator 1 and oscillator 2, and then *Noise* controls the balance between both oscillators and the noise generator.

Next comes the *FILTER* section, which has a high cut (or low-pass) filter for processing output from both the oscillators and the noise generator.

Any noise generator signal is then passed to a low cut (or high-pass) filter. Individual cutoff frequency controls are available for both the *High Cut* and the *Low Cut* filter, and a single *Q* parameter controls resonance for both filters.

The final section offers a control for the instrument's *Vel Sens.*(itivity) and a level control for its *Output*.

Modulation Sources:

› *VEL*(ocity) [monophonic] - The velocity portion of incoming note messages.

Nested Device Chains:

› *FX* - A chain for processing the device's entire audio output.

15.2.5. E-Tom

An electronic tom instrument with optional pitch modulation.

The *GEN* section contains parameters for controlling and processing the instrument's slightly rectified sine oscillator. The frequency of this oscillator is set by the *Tune* knob, and its level is controlled by an AD envelope that has a short, fixed attack time and an exponential, adjustable *Decay* time. The *Click* option adds impact to the sound by doubling portions of it, and the *Tone* control sets the cutoff frequency of a gentle low-pass filter.

The *PEG* section concerns a separate AD envelope generator that controls the oscillator's pitch. You can adjust the *Decay* time, the shape of that decay segment with the contour control, and the *Amount* of modulation in semitones.

The final section offers a control for the instrument's *Vel Sens.*(itivity) and a level control for its *Output*.

Modulation Sources:

› None

Nested Device Chains:

› *FX* - A chain for processing the device's entire audio output.

15.2.6. FM-4

A four-oscillator FM synthesizer with frequencies set as ratios with offsets, optional self-modulation, a noise generator with a resonant low-pass filter, and a modulation matrix. Each row of the matrix represents one of the four oscillators as a modulation destination, and each column is labeled with the modulation source it represents.

On the far left are four indentically equipped sections, representing the four sine oscillator units of the instrument. Oscillator *1* is at top, oscillators *2* and *3* follow, and oscillator *4* is at bottom.

In each unit, the two central controls help determine the sine oscillator's frequency. Each incoming note message is multiplied by the top, unlableled numeric control to set the oscillator's base frequency for that voice. For example, playing a note message of A4 (440Hz) with a setting of *1.00* triggers that oscillator at 440Hz. Playing A4 again with a setting of *2.00* would set the oscillator to 880Hz, just as a setting of *0.50* would tune the oscillator to 220Hz in this example. This system also allows you to see the frequency settings of two oscillators as a ratio, a very handy way of thinking in FM synthesis.

The numeric control at bottom is an offset, allowing you to then detune each oscillator by a number of Hertz.

The *Mod* control at the right of each oscillator unit attenuates the output of the oscillator to all frequency modulation connections (this does not affect the audio output of the oscillator). Similarly, the oscillator number in the left of each unit is a button for enabling/disabling that oscillator for modulation purposes (again, the audio output for each oscillator is unaffected by the setting of this toggle).

To the right of oscillator *1* is the *NOISE (N)* section. This noise generator is configured somewhat similarly to the oscillators, with a global *Mod*(ulation) level control at its far right and a button to enable/disable modulation usage at the far left (shown as *N*).

Between these controls are knobs for the *Cutoff* frequency and *Q* of a low-pass filter that the noise generator is connected to, as well as a *Drive* control that can boost the output signal by up to *+48.0 dB*.

While the *MATRIX* section that follows is somewhat cryptic, it is the heart of the instrument's frequency modulation model. This table shows the individual amounts of modulation between the five generators that we have just discussed. The columns represent the *sources* of modulation, and the rows represent the four oscillator units, which are the potential frequency modulation *destinations*. Similar to early digital FM synthesizers, these signal attenuators go from *0* (no signal/modulation) to *100* (the fullest amount of modulation available). In this sense, you could also think of these gain values as percentages of modulation.

Note

Just remember that the settings in each oscillator and noise generator unit impact the matrix values. Each modulation amount in the grid is scaled by the source's global *Mod*(ulation) level and is completely bypassed if the modulation enable/disable switch is flipped off.

As an example, let's look at the third column, which is labeled *3*. Each of the rows in this column represents one of the respective oscillator units as a destination and the amount of attentuation applied to that particular modulation connection. The first row in this column shows the amount that oscillator *3* modulates the frequency of oscillator *1*. Accordingly, the second, third, and fourth rows control the amount that oscillator *3* modulates the frequencies of oscillators *2*, *3*, and *4*, respectively. In any other column, the destinations would be the same but the source would be either a different oscillator (the numbered columns) or the noise generator unit (column *N*).

And as the example above indicates, oscillator *3* can be set to modulate itself — to effectively "feed back" — by setting the third row attentuator to a value greater than zero. The same is true of all four oscillator units when the output of an oscillator is set to modulate its own matrix input.

The section to the right of the *NOISE (N)* and *MATRIX* sections is the instrument's audio mixer. Each generator unit has an attenuator for setting the amount signal that will reach the instrument's audio output. Just as the matrix and other modulation controls did not affect the audio level of each unit, these gain controls do not affect modulation levels in any way.

The large section that comes next contains the instrument's envelope generators and a global pitch control (along with most of the other unified modulation sources).

The first row of controls belongs to the amplitude envelope generator unit (*AEG*), which affects the entire instrument's audio output level and can also be routed to additional modulation destinations. After the modulation routing button are standard *A*ttack, *D*ecay, *S*ustain, and *R*elease controls, as well as a bipolar control for each voice's *Gain*.

The following three rows provide controls for envelope generators 2 (*EG2*), 3 (*EG3*), and 4 (*EG4*). These three envelope generator units are identical in structure. Following each modulation routing button are standard *A*ttack, *D*ecay, *S*ustain, and *R*elease controls, as well as a global *Depth* control, which attentuates the output of that envelope generator before the signal is passed to its assigned modulation destinations.

At the bottom left of this section is a bipolar global *Pitch* control that is set in semitones. This knob affects the pitch of all four oscillator units and has a range of down one octave (*-12.00*) to up one octave (*12.00*).

The next to last section is dedicated to the instrument's polyphonic LFO (*PLFO*) modulation source. Beneath the modulation routing button is the waveshape selector. This graphical control can be clicked and dragged to toggle between sine, triangle, square, ramp, sawtooth, and random shapes, respectively. On the next row are toggles for the LFO's polarity (*±*) and whether or not the LFO retriggers with each note played (*R*).

The following two controls govern the frequency of the LFO. The drop-down menu sets the time base in which you are working for the numeric control below. If the menu is set to *time_seconds*, then the control below sets the frequency in Hertz. If the menu is set to anything else (*time_beat*, *time_dotted*, or *time_triplets*), then the control below will provide musical, timeline-synced options, such as full bars (*4/1* for a period of four bars) or specific note values (*1/4* for one cycle per quarter note) of the type selected in the menu (either full beats, dotted notes, or triplets of the set denomination).

The *Phase* of the LFO can also be set, as can the global *Depth*, which is applied before the signal is passed to its assigned modulation destinations.

The final section contains three toggle buttons and two knobs. The *MONO* button switches the synthesizer to a monophonic mode, allowing only one note to be played at a time. The next two buttons will take effect only when both *MONO* mode is enabled and you are playing in a legato style. The *ST* button is a "single trigger" option, which prevents envelopes from retriggering. And the *FG* button enables a "fingered glide" mode, which creates portamento between successive notes. Finally, controls for the instrument's *Glide* time (for when the *FG* button is active) and *Output* level are available.

Modulation Sources:

› *AEG* (amplitude envelope generator) [polyphonic] - The signal of this instrument's amplitude envelope generator module. (The routing of this module to the instrument's amplitude is hardwired.)

› *EG2* (envelope generator 2) [polyphonic] - The signal of this instrument's second envelope generator module.

› *EG3* (envelope generator 3) [polyphonic] - The signal of this instrument's third envelope generator module.

› *EG4* (envelope generator 4) [polyphonic] - The signal of this instrument's fourth envelope generator module.

› *VEL*(ocity) [polyphonic] - The velocity value for each voice.

› *KEY* [polyphonic] - The key value played for each voice, including the voice's Micro-Pitch expression (see section 10.1.3).

› *MW* (modulation wheel) [monophonic] - The values of any incoming MIDI messages for continuous controller number 1.

› *AT* (aftertouch) [monophonic] - The values of any incoming MIDI aftertouch (or channel pressure) messages.

› *TMB* (timbre) [polyphonic] - The bipolar output of each voice's timbre expression (see section 10.1.2.4).

› *PLFO* [polyphonic] - The signal of this instrument's polyphonic LFO module.

Nested Device Chains:

› *Note FX* - A chain for processing incoming note messages before they reach this device.

› *FX* - A chain for processing the device's entire audio output.

15.2.7. Organ

A tonewheel organ.

The *DRAWBARS* section contains nine standard gain faders (the vertically higher the fader, the louder the gain), each of which represents the level of the respective drawbar harmonic. In order, these harmonics are:

› Fader 1 - *Sub*, or one octave below the fundamental (in organ notation, 16′ [feet])

› Fader 2 - *5th*, or one fifth above the fundamental (5 1/3′)

› Fader 3 - *Primary*, or the fundamental (8′)

› Fader 4 - *8th*, or one octave above the fundamental (4′)

› Fader 5 - *12th*, or one octave and a fifth above the fundamental (2 2/3′)

› Fader 6 - *15th*, or two octaves above the fundamental (2′)

› Fader 7 - *17th*, or two octaves and a major third above the fundamental (1 3/5′)

› Fader 8 - *19th*, or two octaves and a fifth above the fundamental (1 1/3′)

› Fader 9 - *22nd*, or three octaves above the fundamental (1′)

The final section at bottom offers a control for the instrument's *Output* level.

Modulation Sources:

› *TMB* (timbre) [polyphonic] - The bipolar output of each voice's timbre expression (see section 10.1.2.4).

› *MW* (modulation wheel) [monophonic] - The values of any incoming MIDI messages for continuous controller number 1.

› *AT* (aftertouch) [monophonic] - The values of any incoming MIDI aftertouch (or channel pressure) messages.

Nested Device Chains:

› *Note FX* - A chain for processing incoming note messages before they reach this device.

› *FX* - A chain for processing the device's entire audio output.

15.2.8. Polysynth

A subtractive synthesizer with two highly dynamic oscillators, a noise generator, a multimode filter, and numerous modulation sources.

This instrument starts with two substantial oscillator units. Oscillator 1 is found on top, and oscillator 2 is on bottom. As the oscillators are completely identical in structure and parameters, we will only discuss them once.

At the top of each oscillator unit is a dynamic waveshape display. As oscillator parameters are adjusted, this display will reflect the current waveshape generated by this oscillator.

The *Pitch* of an oscillator can be adjusted by a perfect fifth up or down (from *-7.00 st* [semitones] to *7.00 st*). Below this *Pitch* knob is an octave switch in organ foot notation. From the default setting (*8′*) the oscillator can be set from two octaves down (*32′*) to three octaves up (*1′*), or any octave in between.

The *Shape* control allows you to blend three distinct waves. At the center position, you get only a sawtooth wave at the current pitch. Moving from the center position to the left cross-fades into a pulse wave that is one octave up. Moving from the center position to the right cross-fades into a saw that is one octave up. Below this *Shape* knob is a pulse width control that affects both the pulse wave at the left position and the sawtooth at the right position.

A *Sub* pulse wave that is one octave down can also be blended in. Below this *Sub* level knob is a pulse width control for the sub wave.

Each oscillator unit can also be syncronized to a tunable oscillator. The *Sync* knob controls the frequency of the master sync oscillator as an offset from the oscillator unit's pitch (from *0.00* semitones [unison; no effect] to *60.00* semitones [five octaves up]). The reset button (*R*) beneath the *Sync* knob causes the oscillator unit to return to its initial phase for each incoming note.

Finally, the lower control at the far right determines the number of voices used for each note played by this oscillator unit. Settings range from *1v* (one single voice per note) to *16v* (16 voices per note). When more than one voice is active here, the *Unison* knob above becomes active, allowing you to set the maximum detuning per voice from no detuning (*0 cents*) up to a full semitone (*100 cents*).

The *MIX* section primarily concerns the blend of the instrument's generator units. The *1/2* knob controls the balance between oscillator 1 and oscillator 2. The *Noise* knob then controls the balance between both oscillators and a white noise generator. The bipolar master *Gain* control allows adjustment of the mixer output's level. Finally, the bipolar global *Pitch* control adjusts the frequency of both oscillators. This control is set in semitones, with a range of three octaves in either direction (from *-36.00* to *36.00*).

The instrument's *FILTER* module and accompanying envelope generator are found in the next section. The first control sets the filter's mode. This graphical control at top can toggle between seven filter types: a gentle low-pass filter, a 4-pole low-pass filter, a gentle band-pass filter, a 4-pole band-pass filter, a gentle high-pass filter, a 4-pole high-pass filter, and a band-reject filter, respectively. The next row includes filter controls for the cutoff *Freq*(uency), the *Reso*(nance) amount, the degree to which the cutoff frequency is controlled by incoming *Key*(board) and note pitches, and a control labeled *Osc FM*, which uses an audible-rate oscillator of fixed frequency to modulate the filter's cutoff frequency.

The *AMP* section represents the instrument's main amplifier, which is under the control of a hardwired envelope generator. Provided here are standard *A*ttack, *D*ecay, *S*ustain, and *R*elease controls.

The next three modules represent the instrument's LFO units. While *LFO1* and *LFO2* are both monophonic (meaning there is a single instance of those LFOs that applies an identical signal to each voice), *PLFO* is polyphonic (meaning it can have a separate copy of itself for each voice triggered). Otherwise, the architecture and parameters of all three LFO units are identical.

After each LFO unit's modulation routing button is a control for *Phase* and one for global *Depth*, which is applied before the signal is passed to its assigned modulation destinations. To the right of these controls are toggles for the LFO's polarity (*±*) and whether or not the LFO retriggers with each note played (*R*).

The bottom row of each LFO unit starts with two controls that govern the frequency of the LFO. The drop-down menu sets the time base in which you are working for the numeric control to its right. If the menu is set to *time_seconds*, then the following control sets the frequency in Hertz. If the menu is set to anything else (*time_beat*, *time_dotted*, or *time_triplets*), then the numeric control will provide musical, timeline-synced options, such as full bars (*4/1* for a period of four bars) or specific note values (*1/4* for one cycle per quarter note) of the type selected in the menu (either full beats, dotted notes, or triplets of the set denomination).

The final control in this last row is the waveshape selector. This graphical control can be clicked and dragged to toggle between sine, triangle, square, ramp, sawtooth, and random shapes, respectively.

The final parameter section contains three toggle buttons and four knobs. The *MONO* button switches the synthesizer to a monophonic mode, allowing only one note to be played at a time. The next two buttons will take effect only when both *MONO* mode is enabled and you are playing in a legato style. The *ST* button is a "single trigger" option, which prevents envelopes from retriggering. And the *FG* button enables a "fingered glide" mode, which creates portamento between successive notes. Finally, controls for the instrument's *Glide* time (for when the *FG* button is active), *Vel Sens.*(itivity), *Output* level, and *Pan*(ning) are available.

Modulation Sources:

› *LFO1* [monophonic] - The signal of this instrument's first monophonic LFO module.

› *LFO2* [monophonic] - The signal of this instrument's second monophonic LFO module.

› *PLFO* [polyphonic] - The signal of this instrument's polyphonic LFO module.

› *VEL*(ocity) [polyphonic] - The velocity value for each voice.

› *KEY* [polyphonic] - The key value played for each voice, including the voice's Micro-Pitch expression (see section 10.1.3).

› *TMB* (timbre) [polyphonic] - The bipolar output of each voice's timbre expression (see section 10.1.2.4).

› *FEG* (filter envelope generator) [polyphonic] - The signal of this instrument's filter envelope generator module.

› *AEG* (amplitude envelope generator) [polyphonic] - The signal of this instrument's amplitude envelope generator module. (The routing of this module to the instrument's amplitude is hardwired.)

› *MW* (modulation wheel) [monophonic] - The values of any incoming MIDI messages for continuous controller number 1.

› *AT* (aftertouch) [monophonic] - The values of any incoming MIDI aftertouch (or channel pressure) messages.

Nested Device Chains:

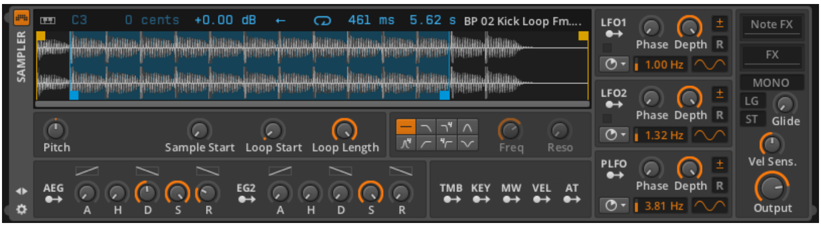
› *Note FX* - A chain for processing incoming note messages before they reach this device.

› *FX* - A chain for processing the device's entire audio output.

15.2.9. Sampler

A sampler that can handle single or multiple samples (with a resizable mapping editor) and has a multimode filter and numerous modulation sources.

This instrument plays back one or more audio files as its source material. The instrument's primary section at the top left concerns the current source material. The options here differ in cases where a single audio file is loaded or when multiple audio files are being used.



When only one audio file is loaded into the instrument, all relevant parameters appear within this section.

All numeric controls and toggle buttons appear across the top row. When a button can be toggled off and on, it appears gray when disabled and blue when activated. Numeric controls are tinted a very faint blue when they have no effect because of other settings.

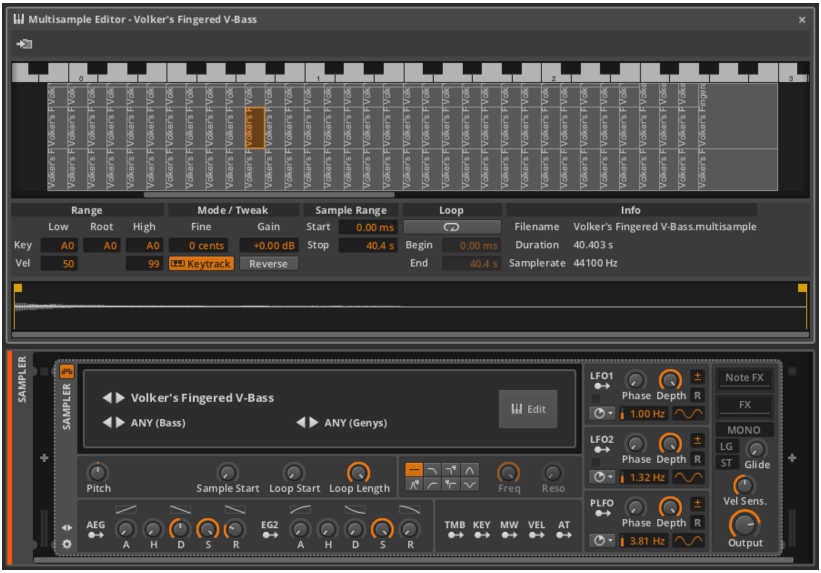
The first button looks like a small piece of a piano keyboard, and it toggles whether keyboard tracking is enabled. When this button is disabled, any note placed will trigger the sample at its original pitch and speed. When it is enabled, the following note number control defines the root key for this sample. The root key is the note where no transposition occurs for the sample, and all other keys shift the sample's pitch and speed by a porportional amount. Having keyboard tracking engaged also enables the following fine tuning control, which can offset the sample's pitch by up to one semitone in either direction (from *-100 cents* to *100 cents*).

The gain control that follows adjusts the sample's level from anywhere between *-12.0 dB* and *12.0 dB*. The left-facing arrow button that comes next enables reverse mode, causing the sample to play backwards, starting at its end and moving towards the beginning.

The oval-shaped arrow button toggles looping off or on for this sample. When looping is enabled, the following two time controls represent the points in the sample where looping begins and ends, respectively.

The white text that appears at the end of the top row simply indicates the name of the sample file that is currently loaded.

Beneath this row of controls is a waveform display for the loaded sample. This display also has two adjustable yellow flags that represent start and stop times for the sample's playback. (These flags are the only way to adjust the sample's start and stop times.) When looping is enabled, two adjustable blue flags that represent the loop's beginning and end points will be displayed as well.



When multiple audio files are loaded into the instrument, this section provides direct access to the Preset, Category, and Creator menus (see section 7.2.2) as well as a large *Edit* button, which loads the instrument's multisample editor into the central panel area, above the **Device Panel**.

The multisample editor has four horizontal layers:

› The keyboard at top provides a reference for sample mappings. It is for display only.

› The grid-like zone layer indicates where sample lie in terms of notes (horizontally) and velocity levels (vertically). To choose a sample for display and manipulation, it must be selected in this zone layer.

› The parameter layer contains all numeric controls and toggle settings for the selected sample.

  The *Range* section on the left contains the sample's *Key* and *Vel*(ocity) settings. The *Low* values set the bottom key and velocity at which the sample will be triggered. Similarly, the *High* values set the top key and velocity at which the sample will be triggered. (The zone layer can also adjust these four settings by clicking and dragging one of the edges of the sample's zone.) The *Root* value defines the key at which no pitch transposition would be applied to the sample.

  The *Mode / Tweak* section contains four settings to alter playback of the sample. *Fine* offsets the sample's pitch by up to one semitone in either direction (from *-100 cents* to *100 cents*). *Gain* adjusts the sample's level from anywhere between *-12.0 dB* and *12.0 dB*. The *Keytrack* toggle enables the instrument to transpose a sample when it is triggered beyond its root note, and the *Reverse* toggle causes the sample to play backwards, starting at its end and moving towards the beginning.

  The *Sample Range* section sets both the *Start* and *Stop* times to be used for playback of this sample.

  The *Loop* section toggles whether this sample will be looped, and if so, the *Begin*(ning) and *End* times for looping are set here.

  Finally, the *Info* section displays the *Filename* of the loaded multisample file, as well as the *Duration* and *Samplerate* of the selected sample.

› The waveform layer displays the selected sample along with adjustable yellow flags for the sample's *Start* and *Stop* times. If looping is enabled, then the *Begin*(ning) and *End* loop times are shown with adjustable blue flags.

All other sections and controls of this instrument are the same, regardless of the number of samples being used.

The section below on the left contains a few relative controls that affect all samples being used. The *Pitch* of each sample can be shifted in semitones, with a range of three octaves in either direction (from *-36.00* to *36.00*). And the *Sample Start*, *Loop Start*, and *Loop Length* of all samples can be scaled here as well on a percentage basis.

The next section on the right contains the instrument's filter module. The first control sets the filter's mode. This graphical control on the left can toggle between eight options: (top row) no filter, a gentle low-pass filter, a 4-pole low-pass filter, a gentle band-pass filter, (bottom row) a 4-pole band-pass filter, a gentle high-pass filter, a 4-pole high-pass filter, and a band-reject filter, respectively. Also included are standard controls for the filter's cutoff *Freq*(uency) and its *Reso*(nance) amount.

The bottom section on the left contains controls for the instrument's two flexible envelope generator modules. The amplitude envelope generator module (*AEG*) affects the entire instrument's audio output level and can also be routed to additional modulation destination. The second envelope generator module (*EG2*) can be freely routed. Structurally, these modules are identical.

Both envelope generators units have a standard *A*ttack, *D*ecay, *S*ustain, and *R*elease controls. Additionally, the attack, decay, and release segments each have shape controls for changing the curves of these timed segments. Finally, a *H*old control sets the time that the envelope pauses at full strength after the completion of the attack segment and before the decay segment begins.

The next three modules represent the instrument's LFO units. While *LFO1* and *LFO2* are both monophonic (meaning there is a single instance of those LFOs that applies an identical signal to each voice), *PLFO* is polyphonic (meaning it can have a separate copy of itself for each voice triggered). Otherwise, the architecture and parameters of all three LFO units are identical.

After each LFO unit's modulation routing button is a control for *Phase* and one for global *Depth*, which is applied before the signal is passed to its assigned modulation destinations. To the right of these controls are toggles for the LFO's polarity (*±*) and whether or not the LFO retriggers with each note played (*R*).

The bottom row of each LFO unit starts with two controls that govern the frequency of the LFO. The drop-down menu sets the time base in which you are working for the numeric control to its right. If the menu is set to *time_seconds*, then the following control sets the frequency in Hertz. If the menu is set to anything else (*time_beat*, *time_dotted*, or *time_triplets*), then the numeric control will provide musical, timeline-synced options, such as full bars (*4/1* for a period of four bars) or specific note values (*1/4* for one cycle per quarter note) of the type selected in the menu (either full beats, dotted notes, or triplets of the set denomination).

The final control in this last row is the waveshape selector. This graphical control can be clicked and dragged to toggle between sine, triangle, square, ramp, sawtooth, and random shapes, respectively.

The final parameter section contains three toggle buttons and three knobs. The *MONO* button switches the synthesizer to a monophonic mode, allowing only one note to be played at a time. The next two buttons will take effect only when both *MONO* mode is enabled and you are playing in a legato style. The *ST* button is a "single trigger" option, which prevents envelopes from retriggering. And the *FG* button enables a "fingered glide" mode, which creates portamento between successive notes. Finally, controls for the instrument's *Glide* time (for when the *FG* button is active), *Vel Sens.*(itivity), and *Output* level are available.

Modulation Sources:

› *AEG* (amplitude envelope generator) [polyphonic] - The signal of this instrument's amplitude envelope generator module. (The routing of this module to the instrument's amplitude is hardwired.)

› *EG2* (filter envelope generator) [polyphonic] - The signal of this instrument's second envelope generator module.

› *TMB* (timbre) [polyphonic] - The bipolar output of each voice's timbre expression (see section 10.1.2.4).

› *KEY* [polyphonic] - The key value played for each voice, including the voice's Micro-Pitch expression (see section 10.1.3).

› *MW* (modulation wheel) [monophonic] - The values of any incoming MIDI messages for continuous controller number 1.

› *VEL*(ocity) [polyphonic] - The velocity value for each voice.

› *AT* (aftertouch) [monophonic] - The values of any incoming MIDI aftertouch (or channel pressure) messages.

› *LFO1* [monophonic] - The signal of this instrument's first monophonic LFO module.

› *LFO2* [monophonic] - The signal of this instrument's second monophonic LFO module.

› *PLFO* [polyphonic] - The signal of this instrument's polyphonic LFO module.

Nested Device Chains:

› *Note FX* - A chain for processing incoming note messages before they reach this device.

› *FX* - A chain for processing the device's entire audio output.

15.3. Containers

Each *container* is a device whose primarily function is hosting other devices.

As each container has a different purpose, the primary signal I/O is listed for each device. (For more information, see section 14.1.2.)

15.3.1. Drum Machine

(Notes in, Audio out) A container that routes note signals to specific chains based on their pitch. Each chain has its own internal mixer controls. (For more information, see section 14.1.2.1.)

15.3.2. FX Chain

(Audio in, Audio Out) A container that houses a serial audio device chain.

15.3.3. FX Layer

(Audio in, Audio out) A container that houses parallel audio chains. Each chain has its own internal mixer controls. (For more information, see section 14.1.2.3.)

15.3.4. Instrument Chain

(Notes in, Audio out) A container that is made for an instrument and its signal chain. Controls are provided for offsetting the chain's output gain and for attenuating the chain's output level. Any incoming audio that reaches this device is passed through without attenuation.

15.3.5. Instrument Layer

(Notes in, Audio out) A container that houses multiple instruments in parallel. Each chain has its own internal mixer controls. (For more information, see section 14.1.2.2.)

15.3.6. Mid-Side Split

(Audio in, Audio out) A container that takes a normal stereo signal and splits it into its mid (centered) and side (panned) components, each of which is provided with an independent chain.

15.3.7. Multiband FX-2

(Audio in, Audio out) A container that splits the incoming audio at a definable frequency and provides independent chains for the audio below and above that frequency.

15.3.8. Multiband FX-3

(Audio in, Audio out) A container that splits the incoming audio at two definable frequencies and provides independent chains for the audio below the first frequency, the audio between the two frequencies, and the audio above the second frequency.

15.3.9. Replacer

(Audio in, Audio out) A container that filters and analyzes the level of the incoming audio signal, and when the signal rises above a set threshold, notes are generated at a set pitch and velocity. These notes and the original (dry) audio signal are then passed to the internal *INST* (for instrument) device chain.

15.3.10. XY Effect

(Audio in, Audio out) A container that loads up to four audio FX in parallel and allows you to crossfade their outputs.

15.3.11. XY Instrument

(Notes in, Audio out) A container that loads up to four instruments in parallel and allows you to crossfade their outputs.

15.4. Audio FX

Each *audio FX* device manipulates the incoming audio signal before passing it onward.

15.4.1. Bit-8

An audio degrader that affects the perceived bit depth and clocking.

15.4.2. Blur

A comb-filter diffusion effect where each channel has two comb filters, each with a feedback control.

15.4.3. Chorus

A chorus effect with an adjustable LFO with phase offset for the right channel (*R Phase*).

15.4.4. Comb

A comb filter effect with frequency and bipolar feedback controls.

15.4.5. Compressor

A compressor with standard threshold, ratio, gain, and timing settings.

15.4.6. De-Esser

A de-esser with a variable high-pass filter and monitoring option for the detection circuit.

15.4.7. Delay-1

A tempo-syncable delay with uniform delay time, offset, and feedback settings for the left and right channels.

15.4.8. Delay-2

A tempo-syncable delay with discrete delay time, offset, and feedback settings for the left and right channels. This device also has warble (*Detune* and *Rate*) and *Crossfeed*(back) settings.

15.4.9. Distortion

A distortion effect based on hard clipping with a peak EQ before the clipping is applied and high- and low-pass filters after.

15.4.10. Dynamics

A flexible dynamics processor that allows for either downward or upward compression on both the loud and quiet parts of the sound. The device also has a sidechain input, an *FX* device chain for the control signal, and a graphical interface.

15.4.11. EQ-2

A two-band parametric equalizer with resonant filter modes and a graphical interface.

15.4.12. EQ-5

A five-band parametric equalizer with resonant filter modes and a graphical interface. The device also has global controls to morph the strength (*Amount*) and placement (*Shift*) of the EQ curve.

15.4.13. EQ-DJ

A three-band equalizer with definable crossover frequencies and mute controls for each band.

15.4.14. Filter

A multimode filter with pre- and post-gain.

15.4.15. Flanger

A flanger effect with an adjustable LFO and feedback parameters for both magnitude (*Feedb.*) and phase (*Neg.*). This device can be set to *Retrig*(ger) on incoming note messages.

15.4.16. Freq Shifter

A frequency shifter with an adjustable frequency range. This device can also distribute the upward and downward frequency shift across the stereo field.

15.4.17. Gate

A noise gate with sidechain input and an *FX* device chain for the control signal.

15.4.18. Ladder

A multi-mode ladder filter with a built-in LFO, envelope, and envelope follower to modulate the filter's frequency.

15.4.19. Peak Limiter

A limiter with peak level, gain, and release controls.

15.4.20. Resonator Bank

A bank of six resonant filters that have frequency, resonance, and gain controls. The device also has global controls to morph these three controls as well as keyboard tracking to offset the filters' frequencies based on incoming note signals.

15.4.21. Reverb

An algorithmic reverb effect with distinct controls for *EARLY* reflections and for the later dense reflections (*TANK*). The *TANK* is split into three assignable bands with relative delay times for the low and high bands. This device also has a graphical interface.

15.4.22. Ring-Mod

A ring modulator with a definable frequency and a *Mix* control for blending the source material with the resultant sum and difference tones. The device also has *Pre*- and *Post*-processing device chains.

15.4.23. Rotary

A rotary-speaker emulation that modulates the signal's placement in the stereo field.

15.4.24. Tool

A utility tool for signals that includes amplitude, panning, and width controls as well as channel invert switches and high-resolution output level meters.

15.4.25. Transient Control

A transient detector that can make onsets and sustain segments relatively louder or softer.

15.4.26. Tremolo

An amplitude modulator that is controlled by an LFO of various waveshapes. This device can be set to *Retrig*(ger) on incoming note messages.

15.5. Generators

Each *generator* device freely outputs signal without the need for any input.

15.5.1. Test Tone

A generator that outputs a sine wave at a set frequency and level.

15.6. Modulators

Each *modulator* device can modulate the parameters of any device found within its own *FX* device chain. In terms of signal flow, a modulator passes the note and audio signals it receives directly to its outputs and to the *FX* device chain.

15.6.1. Audio MOD

A modulator that applies a filter and envelope follower to an incoming audio signal, which is then used as the control signal.

15.6.2. LFO MOD

A modulator that provides two low-frequency, tempo-syncable oscillators as independent modulation sources.

15.6.3. Note MOD

A modulator that takes incoming or designated note signals and creates summed, monophonic versions of their expressions along with a configurable envelope signal.

15.6.4. Step MOD

A step sequencer whose output is used as a modulation source.

15.7. Routers

Each *router* device allows the redirecting of a track's signal path. To achieve this, a router often contains audio and/or note chooser menus for addressing an incoming or outgoing signal to the appropriate destination, including destinations outside of Bitwig Studio.

As each router has a different purpose, the primary signal I/O is listed for each device.

15.7.1. Audio Receiver

(Audio in, Audio out) A router that imports audio signal from any designated project source.

15.7.2. Hardware FX

(Audio in, Audio out) A router that sends the incoming stereo audio signal out of the track and system, and then returns another stereo signal back.

15.7.3. Hardware Instrument

(Notes in, Audio out) A router that sends the incoming note signals out of the track and system, and then returns the resultant audio.

15.7.4. Note Receiver

(Notes in, Notes out) A router that imports note signals from any designated project source.