

# audial manipulators

## About

I love music. I love video games. And I love things that are cheap. The *Audial Manipulators* package is my attempt at blending all of these together. Unity offers several audio filters in its Pro version but, for many of us, that's outside of budgetary constraints (my budget is currently \$0). I wanted to design tools that interested me and, as the project grew, I quickly came to the realization that it would be beneficial for others as well.

The Audial Manipulators asset package is designed to give you expressive control over your audio from within the Unity engine. Each component is intended to be functional regardless of what platform you are exporting to or whether you are using the Free or Pro version of Unity. Included in this pack are 11 effects, 1 editor utility, and 2 demo scenes.

Each Audial Manipulator can be added as a component to any Game Object that has either an Audio Source or an Audio Listener. The effects are designed to be used in tandem with your current audio solutions and are capable of powerful manipulation of your existing soundscapes. All components can be configured in the inspector or manipulated through code for in-game manipulation.

I would love to see how you use this package and am completely open to developing further components based off of user needs. Please let me know if you run into any issues and feel free to contact me with any feedback, suggestions, or promises of cupcakes.

Many thanks to the people who tested this out and gave input, I've really learned a lot from all of you. In particular I'd like to thank to rxi (<https://github.com/rxi>) for his expertise and encouragement along the way. These components wouldn't be very functional if it wasn't for his guidance. Thanks also to Local Minimum ([https://twitter.com/Local\\_Minimum](https://twitter.com/Local_Minimum)) whose feedback was incredibly valuable and who is completely responsible for the inclusion of the Audio Tester component.

Thanks,

John Pennington

Email: [atmospherium@gmail.com](mailto:atmospherium@gmail.com)

Twitter: @atmospherium

# Audial Manipulators Overview

## Signal flow

The sequence for audio processing is as follows:

- 1) Audio Sources (if they exist) are processed first, regardless of order.
- 2) Audial Manipulators are processed sequentially in the order that they appear in the Inspector panel.
- 3) Audio Listeners (if they exist) are processed last, regardless of order.

## Tips

Experiment, experiment, experiment. Most of the devices included in this package have a few intended purposes. But just because there's an intended way to use the device, doesn't mean that it's the only way.

Each of the Audial Manipulators is controllable via code. Configure the devices to sound the way you want, but don't forget to let the devices respond to the player's action and movement. Simple automation can go a long way.

Certain components work best in conjunction with other components. If you have 3D audio, running a Reverb component into a Stereo Widener component will give you much better results than using either one alone. Running a Distortion component might make your audio sound too bright if you don't run it through a State Variable Filter set to Low Pass (or maybe that's exactly what you want). Find out what combinations work for you.

## Cautions

Like any other components in Unity, the performance of these devices will vary depending on the computer that runs them. Don't throw thousands of devices on a scene without investigating the impact that they will have.

# Audio Tester (Editor Component)

## Overview

Unity typically only plays audio when you have started playing a scene in Play mode. This can be problematic for managing your audio, as any changes made during Play mode are lost. The Audio Tester is designed to help you overcome this hurdle. This component allows you to modify the settings for all Manipulators in the chain without having to start Play mode, allowing Unity to retain all edits that you make. For editing purposes, this device is *essential* to getting the sounds you want without any hassle. You can play your Audio Source directly from the Inspector Panel and have your Manipulator chain process this signal – all while you can make live edits to the Component Settings.

## Usage

When you first attach the Audio Tester to a Game Object, one of two messages will appear: “AUDIO CLIP REQUIRED FOR TESTING” or “READY FOR TESTING.” If you intend to use this component for its primary purpose (the testing of your audio chain), a status of “READY FOR TESTING” is required. If you are not seeing this message, simply add an Audio Source that contains an Audio Clip and you are ready to go.

Once you are getting the “READY” message, PLAY and STOP buttons should appear. These buttons simply control the Audio Clip that you have configured on your Audio Source. You can add any Audial Manipulators to the Game Object, hit play, and then tweak the settings on your components to suit your needs.

Additionally, this component contains a “Run Effects In Edit Mode” checkbox. This is useful in a couple of ways. Most substantially, this can act like a quick bypass of all effects in the chain while you are in Edit mode (this won’t impact their performance in Testing mode or after compilation). This is great for assessing just how much the Manipulators are impacting the audio signal. Because this feature allows the audio to be bypassed, it also frees up any processing power that would have been taken by the audio filter. When not actively testing, it’s a good idea to uncheck this box.

## Notes

Due to how Unity handles its Audio Buffer, only components that were placed on the Game Object before hitting PLAY will be processed. If you add a new component, simply hit PLAY again and it will be added to the signal chain.

# Manipulator Components

## *Mixing Components (Fader, Pan, Stereo Widener)*

### Fader

The Fader is a simple volume control device, but has a significant advantage over the Audio Source volume controller – it can be placed anywhere in your audio chain. Several Manipulator devices are threshold dependent, and adjusting the Audio Source volume level would result in unintended side effects. Adding the Fader component later in the chain gives you volume control without negatively impacting the sound. Likewise, putting the Fader after a Delay or Reverb component gives a much more reliable result than modifying the Audio Source volume.

### Pan

The Pan component gives you direct control over the stereo spacing of your audio. This device is best used with 2D audio and can be manipulated through code to interesting effect. Like the Fader device, this device duplicates controls that are present in the Audio Source but allows you to more precisely control your audio based on where you place it in the chain.

### Stereo Widener

The Stereo Widener is a unique component in this set and its uses may not be readily apparent, but creative use of it can make your audio much more engrossing. The Stereo Widener takes your stereo audio information and “pushes” it beyond the usual stereo field. This is very useful for making the Delay or Reverb more pronounced in the audio mix. Additionally, this component allows you to make your stereo mix into a mono mix, make it an extra-wide mix, or fade anywhere between the two.

## *Distortion (Crusher, Distortion, Foldback Distortion)*

### Crusher

The Crusher is a standard bitcrusher component that allows you to reduce the bit depth and sample rate of your audio. This effect can be useful for mimicking the sound of 8-bit era music, degrading any speech to sound digital, or a wide range of crazy, glitchy effects. This effect sounds great on any percussion elements.

### Distortion

The standard Distortion component takes your audio and gives it some bite. The way in which this device processes your audio is fairly destructive of the dynamic range, so it is highly recommended to experiment with the Dry/Wet parameter to find the right balance for your need.

### Foldback Distortion

The Foldback Distortion differs from the standard Distortion component in how it handles the dynamic range of the audio it is affecting. The Foldback Distortion retains a lot more dynamic range and can be a lot more forgiving. However, experimenting with the Threshold parameter can result in some extreme distortion sounds. This device is incredibly versatile and is worth taking time to experiment with.

## *Multi-purpose (Delay, Reverb, State Variable Filter)*

### Delay

The Delay device is an extremely useful device in a wide range of scenarios. Using basic Delay parameters make you audio sound like it's echoing off of walls. Using the Ping-Pong feature on a fast, widely-panned delay can make your mono tracks feel like they are in stereo space. Using this device in conjunction with the Reverb component is incredibly useful for making your audio feel more organic.

### Reverb

The Reverb device approximates the natural reverberation that happens to sound in real life. This is useful for making your sound feel more natural and giving a sense of space to your audio. I highly recommend using code to automate the Dry/Wet mix based on a player's distance from the audio source.

### State Variable Filter

The State Variable Filter isn't just an effect, it's a collection of audio manipulation utilities in its own right. Whether you want to make your audio sound thin or full, duller or brighter, natural or synthetic – it doesn't matter. Whatever your goals, the State Variable Filter is useful in just about any circumstance. Given its usefulness and wide range of uses, there is a section of documentation dedicated solely to exploring its use (please see page 6).

## *Compression (Compressor, Saturator)*

### Compressor

The Compressor shrinks the dynamic range of your audio, evening out and blending any louder elements of your audio. This device isn't restricted to this purpose, however, creative usage of the Attack and Release values can result in some unique audio manipulation.

### Saturator

The Saturator is a hybrid compression/distortion device that takes your audio and simultaneously evens out its dynamic range and brightens its sound. The louder the audio gets, the more the saturator distorts it. This is incredibly useful for squashing your audio together in a way that gives it a bit of a bite.

# State Variable Filter

## Overview

The State Variable Filter is incredibly powerful and is capable of a wide range of manipulative possibilities. There are 6 different Filters included in this Component and each fall into one of two categories – exclusive or additive.

## Exclusive Filters

The exclusive settings for the State Variable Filter are as follows: Low Pass, High Pass, and Band Pass. Each of these Filters affect audio in an all-or-nothing kind of way - if you add a Band Pass, only audio within that frequency range will come through.

- Low Pass – Only audio information below the specified frequency is allowed through. This filter cuts out any of the higher frequencies that make audio sound clearer. This is perfect for making audio feel further away or emphasizing the lower rumble of your sound effects.
- High Pass – Only audio information above the specified frequency is allowed through. Using this effect on any radio chatter or speakers will go a long way in making them feel authentic.
- Band Pass – Only audio information near the specified frequency is allowed through. This is great for “boxing in” your audio and accenting just the frequencies that suit your needs.

## Additive Filters

The additive settings for the State Variable Filter are as follows: Low Add, High Add, and Band Add. Like their exclusive counterparts, these Filters affect frequencies within specific ranges. However, unlike the exclusive settings, the original audio signal is retained and the filtering is added to it (controlled by the Additive Gain setting). A positive value adds information to the signal while a negative value subtracts it.

- Low Shelf – Audio information below the specified frequency is duplicated and added to the original signal. When used with positive gain, this can provide a quick and easy bass boost to your audio. When used with negative gain, this can clean up some of the muddiness of your audio without thinning it out too much.
- High Shelf – Audio information above the specified frequency is duplicated and added to the original signal. When used with positive gain, this can boost the clarity of your signal (particularly if used on vocal assets). When used with negative gain, this can soften your audio and tone down a bit of the harshness that can come through on distorted signals.
- Band Add – Audio information near the specified frequency is duplicated and added to the original signal. When used with positive gain, this can accent whatever frequency ranges you feel need a bit more focus. When used with negative gain, this can reduce the strength of overpowering frequencies without being quite as extreme as the Notch filter.

# Code Usage

## Overview

All components are capable of being configured within the inspector panel, but are also completely manipulable in code. Each script in this package is a part of the “Audial” namespace. The following code demonstrates how to access a Crusher component and modify its Sample Rate property.

```
Audial.Crusher crusher = GetComponent<Audial.Crusher>();  
crusher.SampleRate = 0.025f;
```

Adding new Audial components to a Game Object is just as easy, but does come with a word of caution: When an Audio Source is played, Unity detects all audio filters currently available and will use those filters (and only those filters) until the next time the audio is played. As a result, it is a good idea to make sure that you retrigger your audio after adding any new components. The following code demonstrates this process.

```
Audial.Crusher crusher = gameObject.AddComponent<Audial.Crusher>();  
crusher.SampleRate = 0.025f;  
audio.Play();
```

A few considerations to keep in mind when working with these components via code:

- Instead of adding Components via script, it’s usually a better bet to add the Audial Manipulators you’ll be using and then enabling/disabling them via code. This allows them to be handled more reliably by Unity’s audio manager.
- Since each parameter can be code-controlled, all numeric values are only publicly modifiable through Properties that clamp their values within acceptable ranges. Sending out-of-range value will not cause instability, but may yield different results than you expected. It is best to know what the target ranges are.
- Certain Property accessors run additional processing when utilized. In most instances, these additional processes will not impact either the sound or the performance of the component. There are a few exceptions (see the Delay component for more information).

# Component Properties

The following lists show the full set of Audial Manipulator properties that are code-controllable. Clamped ranges are included (where applicable). Capitalization and spacing are indicative of code usage.

## Compressor

- InputGain – float Range(0,3)
- Threshold – float Range(0,1)
  - Determines the point at which audio compression starts.
- Slope – float Range(0,2)
  - Determines the severity of compression. Higher values = more compression
- Attack – float Range(0.0001f,5)
  - The speed at which the audio is affected.
    - Longer Attack emphasizes audio peaks
    - Shorter Attack minimizes extreme peaks
- Release – float Range(0.0001f,5)
  - The speed at which the compressed signal returns to normal
    - Longer Release retains more of the dynamics of the audio
    - Shorter Release minimizes the difference between soft and loud audio
- OutputGain – float Range(0,5)

## Crusher

- BitDepth – int Range(1,32)
  - Modifies the amount of information per sample (32-bit is standard, 8-bit is NES era)
- SampleRate – float Range(0.001f,1)
  - Update speed of the sample data. Lower = slower
- DryWet – float Range(0,1)

## Delay

- BPM – float Range(40,300) \*RESETS DELAY BUFFER\*
  - Beats Per Minute, used to calculate delay time
- Delay Count – int Range (1,8) \*RESETS DELAY BUFFER\*
  - Number of Delay Units, used to calculate delay time
- Delay Units – int Range (1,32) \*RESETS DELAY BUFFER\*
  - Delay Unit length (whole note to 32<sup>nd</sup> note)
- DryWet – float Range(0,1)
- DecayLength – float Range(0.1f,1);
  - How long the delay signal persists (0.1f is short, 1 is infinite)
- Pan – float Range(-1,1)
  - Affects the panning of the wet signal
  - Dry signal is unaffected
- PingPong – Boolean
  - If signal is panned, subsequent echoes ping pong between speakers



## Distortion

- InputGain – float Range(0,3)
- Threshold – float Range(0.00001f,1)
  - Clipping point for the distortion
  - Impacts the tone/character of the distortion
- DryWet – float Range(0,1)
- OutputGain – float Range(0,5)

## Fader

- Gain – float Range(0,3)
  - Controls volume output
- Mute – Boolean
  - Enables/disables audio output

## Foldback Distortion

- InputGain – float Range(0,3)
- SoftDistortAmount – float Range(0,1)
  - Dry/Wet value for full signal distortion
  - The output from the Soft Distort becomes the input for the primary distortion phase
- Threshold – float Range(0.000001f,1)
  - Threshold for harsher foldback
  - Defines the aggression of the distortion
- DistortAmount – float Range(0,1)
  - Dry/Wet value for primary distortion phase
- OutputGain – float Range(0,5)

## Pan Control

- PanAmount – float Range(-1,1)
  - Pans range from left (-1) to right (1)
  - Defaults to center (0)

## Reverb

- ReverbTime – float Range(0.5f,10)
  - Affects the length of the reverb time
- DryWet – float Range(0,1)

## Saturator

- Amount – float Range(0,1)
  - Controls saturation threshold
  - Also modifies distortion/compression severity

## State Variable Filter

- Frequency – float Range(50,12000)
  - Target frequency

- Resonance – float Range(0,1)
  - Resonance/Quality (Q) of the frequency band
- Drive – float Range(0,0.1f)
  - Drive of the frequency band
- Filter – enum(FilterState)
  - Bypass – no processing
  - Low Pass – only frequencies below the target are retained
  - Low Shelf – frequencies below the target are added or subtracted from signal
  - High Pass – only frequencies above the target are retained
  - High Shelf – frequencies above the target are added or subtracted from signal
  - Band Pass – only frequencies near target are retained
  - Band Add – frequencies near target are added or subtracted from signal
- AdditiveGain – float Range(-1,1)
  - Determines the amount of addition/subtraction of any additive processes
  - Only impacts Low Shelf, High Shelf, and Band Add

### Stereo Widener

- Width – float Range(0,2)
  - Widens the stereo field
  - Ranges from 0 (mono) to 2 (extra-wide)

# Demo Scenes

## Song Mix

This scene demonstrates the basic usage of Audial Manipulators from within the Inspector window. Each instrument has its own chain of effects that are used to expand the sound, whether it's Delay on the drums to add backbeat or Foldback on the distortion on the bass to give some aggression to the mix. The Audio Listener gameObject takes this a step further by mixing all of the audio sources together with a boost from the Compressor and the optional audio degradation from the Crusher.

## LightPong

This scene demonstrates the versatility of modifying your audio via code. There is a single audio clip (lasting just 0.25 seconds) that generates all of the audio in the scene. State Variable Filters are manipulated by any Y-axis movement by the player or the ball, reverbs are modified by the distance between the ball and the paddle, and delays tie the whole thing together. Check out the AudioProcessor script included in the demo's Scripts folder.

# Release Notes

Version 1.0

Initial release.