

Developer

/ Documentation

/ Unreal Engine ▾

/ Unreal Engine 5.4 Documentation

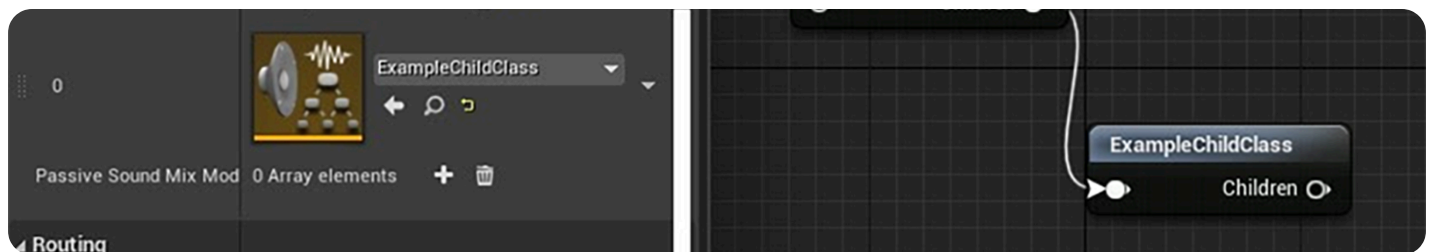
/ Working with Audio

/ Audio Mixing

/ Sound Classes

Sound Classes

An overview of Sound Classes in Unreal Engine.

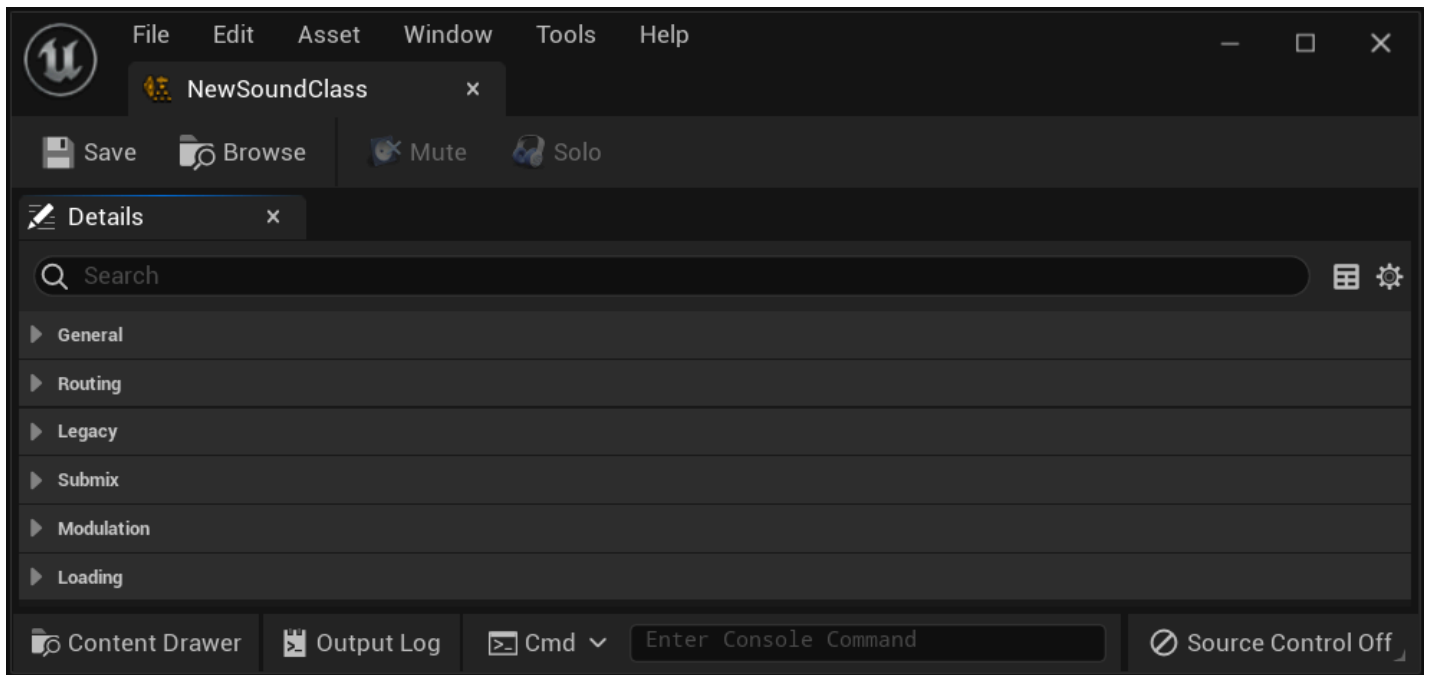


A **Sound Class** is an asset type for the audio engine within **Unreal Engine** that a user can use to group multiple sounds together, then alter the parameters of all relevant Sound Waves at once.

By using **Sound Class Mixes**, many Sound Class parameters can be adjusted during gameplay.

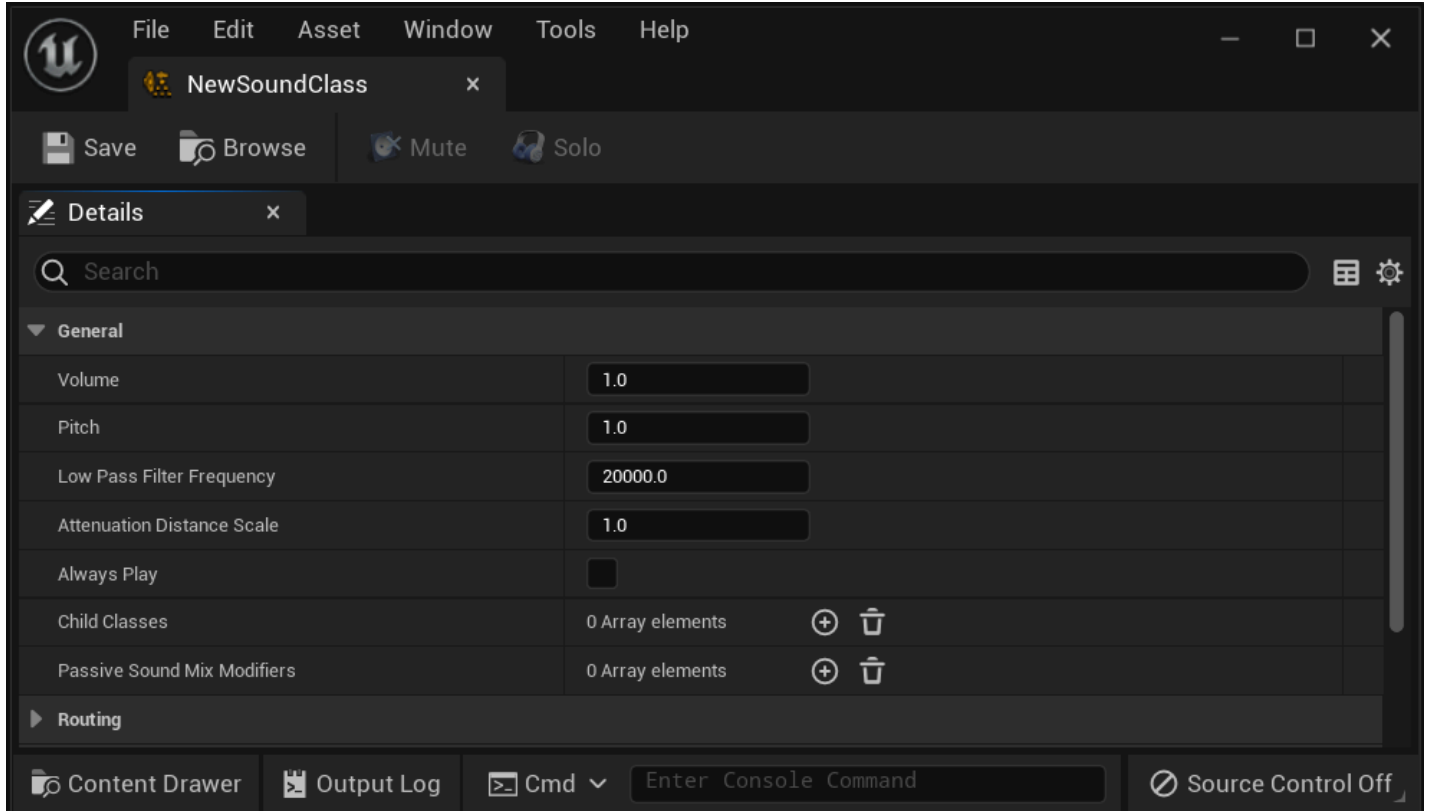
To set up a new Sound Class, right-click from the **Content Browser**, then select **Audio > Classes > Sound Class**.

To open the **Details** panel for a Sound Class, double-click the Sound Class in the Content Browser.



General Properties

The first set of parameters in a Sound Class are the General Properties.



Volume: A multiplier on the loudness of each sound within the Class. This acts on top of Volume settings in the Sound Wave object itself.

Pitch: A multiplier on the pitch of each sound within the Class that acts on top of pitch settings in the SoundWave object.

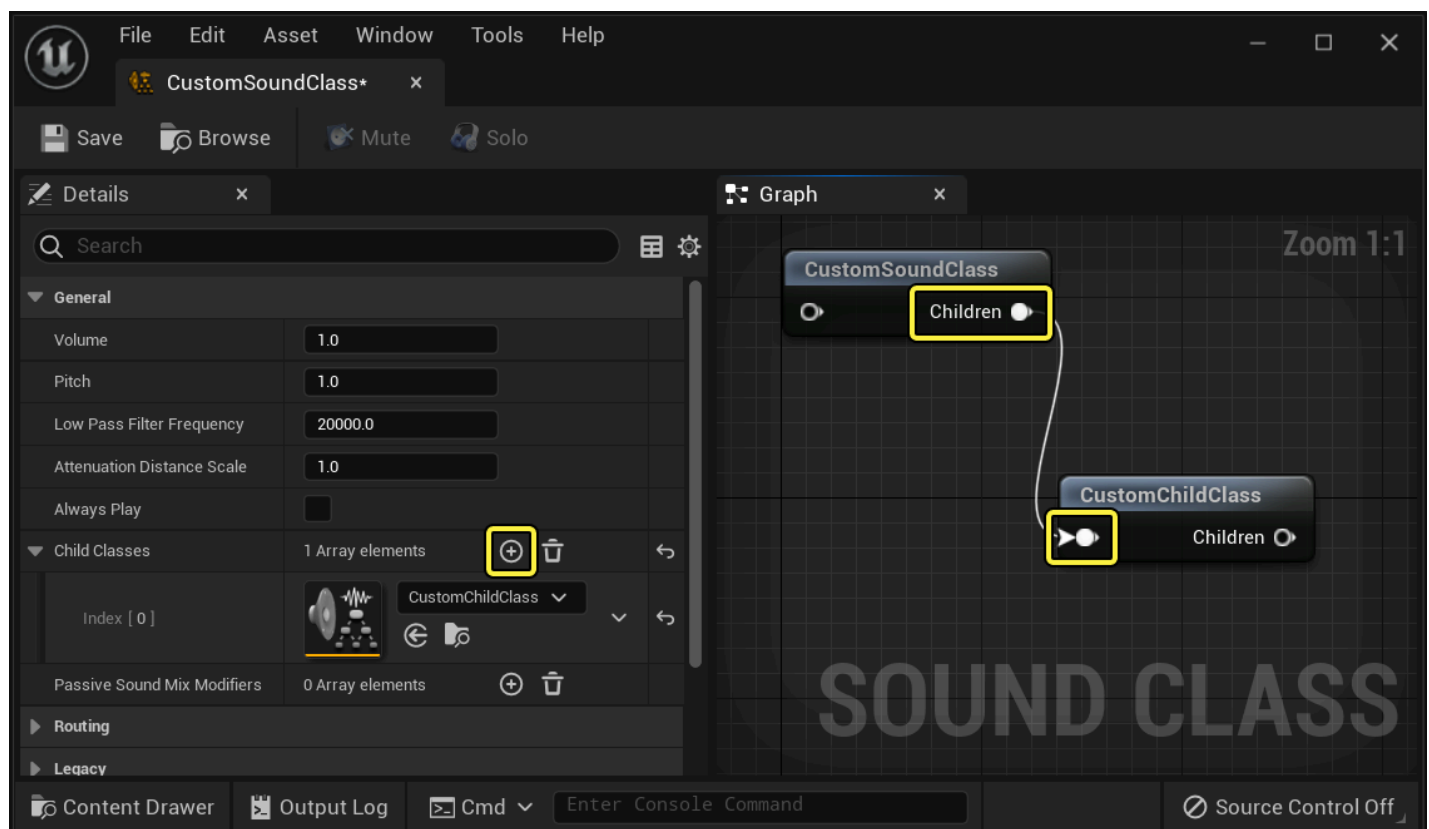
Low Pass Filter Frequency: The cutoff frequency for the low-pass filter that impacts all sounds within the Class; a value of 20,000 or above has no effect on a Sound Wave.

Attenuation Distance Scale: Scales the attenuation distance of each sound within the Class, allowing for attenuation settings to be adjusted dynamically.

Always Play: When enabled, it prevents sounds within the Sound Class from being ejected from the voice pool by sounds with a priority lower than **Always**.

Child Classes

A Sound Class can have other classes set as its **Child Classes**.



To add a Child Class, either click the **plus (+)** icon in line with **Child Classes** and select it from the dropdown, or drag the Sound Class onto the graph and connect the **Children** output pin from the Parent Class node to the input pin of the Child Class node.



When a Sound Class is set as a child, if set to **Inherited** (see [Loading](#)), it will inherit features like Pitch, Volume, and Loading behavior from the Parent Class. In this way, the user can alter

the settings of both the parent and any of its child classes at the same time.

Passive Sound Mix Modifiers

A Sound Class can have an array of **Sound Class Mixes** attached that are triggered automatically when any sound within that Sound Class is played. These are called *passive* because they do not need to be manually activated.

For example, they are triggered automatically when a member of the Sound Class is played, without the need for an explicit **Push Sound Mix Modifier** command. You can still call the Sound Mix Modifier through a **Push** command if you want, and multiple Sound Classes or Blueprints can refer to the same Sound Mix Modifier.

▼ Passive Sound Mix Modifiers	2 Array elements	⊕ ⊖	↶
▼ Index [0]	3 members	▼	↶
Sound Mix	 Dialog_Duckall_PSM	↶ ↷	↶
Min Volume Threshold	0.0		↶
Max Volume Threshold	10.0		↶
▼ Index [1]	3 members	▼	↶
Sound Mix	 Duck_Normal_Attack_MixMod	↶ ↷	↶
Min Volume Threshold	0.0		↶
Max Volume Threshold	10.0		↶

Each Passive Sound Mix Modifier can be given a **minimum** and **maximum volume threshold** for parametric volume to control when it should activate. As an example, this gives the user a way to prevent a far-away sound that has attenuated to an inaudibly quiet volume from triggering a Sound Mix Modifier.

With this feature, a sound designer can create complex behavior in EQ and relative volumes of Sound Classes without adding a lot of Blueprint triggers.

Routing

These settings control how the Sound Class audio is routed through directional channels, with surround-sound setups in mind.

▼ Routing		
LFE Bleed	<input type="text" value="0.0"/>	
Voice Center Channel Volume	<input type="text" value="0.0"/>	
Center Channel Only	<input type="checkbox"/>	
Apply Ambient Volumes	<input type="checkbox"/>	

LFE Bleed: The proportion of the signal that should be sent to the Low Frequency Effects (LFE) Channel. The default is .5, or 50 percent.

Voice Center Channel Volume: The proportion of the signal to be sent to the Center Channel. This is not a multiplier, and the value is not inherited by children.

Center Channel Only: When enabled, the sound will be sent only to the Center Channel.

Apply Ambient Volume: When enabled, the sounds within the Sound Class will be affected by Ambient Zone settings within Audio Volumes, such as Interior/Exterior Volume and Low Pass Filter settings.

Legacy

This tab contains settings that were created before the [Audio Mixer](#) feature was added to the Unreal Audio Engine.

▼ Legacy		
Output Target	Speaker ▼	
▼ Advanced		
Radio Filter Volume	0.0	
Radio Filter Volume Threshold	0.0	
Output to Master EQ Submix	<input type="checkbox"/>	
Is UISound	<input type="checkbox"/>	
Is Music	<input type="checkbox"/>	

Output Target: Changes whether sounds within the Sound Class are sent to the main speaker, an external controller, or a controller (with playback to speaker if no controller present).

Radio Filter Volume: The volume of the radio filter effect that applies to the sounds within the Sound Class. Available on Windows and Xbox with the XAudio2 backend.

Radio Filter Volume Threshold: The volume at which the radio filter kicks in. Available on Windows and Xbox with the XAudio2 backend.

Output to Master EQ Submix: If enabled, sends the sounds within the output for this Sound Class to the reverb system that was used before the Audio Mixer reverb system replaced it. This setting is still respected by the Audio Mixer.

Is UISound: If enabled, the sounds within the Sound Class are audible within UI menus and during regular gameplay. This setting is still respected by the Audio Mixer.

Is Music: If enabled, the sound is marked as music.

Submix

Sounds within a Sound Class can be sent to a **Submix**, which allows multiple sound types to be processed with submix effects at once.

▼ Submix		
Send to Master Reverb Submix	<input checked="" type="checkbox"/>	
Default 2DReverb Send Amount	<input type="text" value="0.0"/>	
Default Submix	<div>None</div> <div>None ▼</div> <div>↶ ↷</div>	

Send to Master Reverb Send Amount: When enabled, primary reverb settings are applied to sounds within the Sound Class.

Default 2D Reverb: The default amount of reverb effect to be placed on non-spatialized sounds.

Default Submix: The submix that the output of sounds within the Sound Class are sent to by default. When set to **None**, the output will be sent to the Sound Class Default Submix specified in Project Settings.

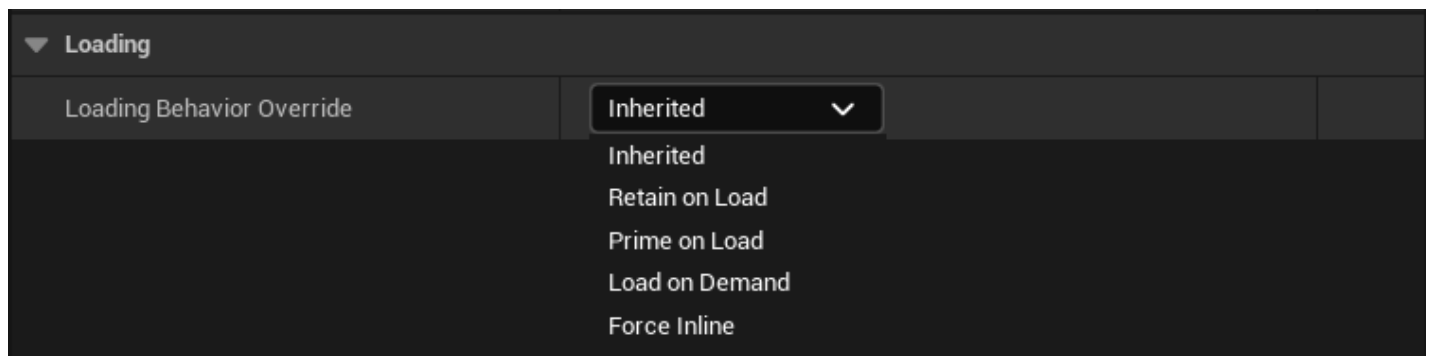
Modulation

If the [Audio Modulation plugin](#) is installed, it is possible to add an array of Modulations Settings to all sounds within a Sound Class.

▼ Modulation		
Volume	<input type="text" value="0.0"/> dB <input type="checkbox"/> Modulate	
Pitch	<input type="text" value="0.0"/> Semitones <input type="checkbox"/> Modulate	
Highpass	<input type="text" value="20.0"/> Hz <input type="checkbox"/> Modulate	
Lowpass	<input type="text" value="20000.0"/> Hz <input type="checkbox"/> Modulate	

Loading

This setting controls how the compressed audio data is loaded.



Inherited: When set to this option, the Sound Class inherits its loading behavior from its parent Sound Class.

Retain on Load: The first chunk of audio for sounds within the Sound Class remains in the cache until explicitly deleted.

Prime on Load: After being loaded, sounds within the Sound Class remain in the cache, but risk being removed to make space for other audio if unused for long periods of time.

Load on Demand: No chunks of the audio are stored until the asset is played or primed.

Force Inline: Forces the sounds within the Sound Class to only use non-streaming decode paths, and no portion of audio is stored in the cache. Requires each Sound Wave asset's Loading Behavior Override to also be set to Force Inline.