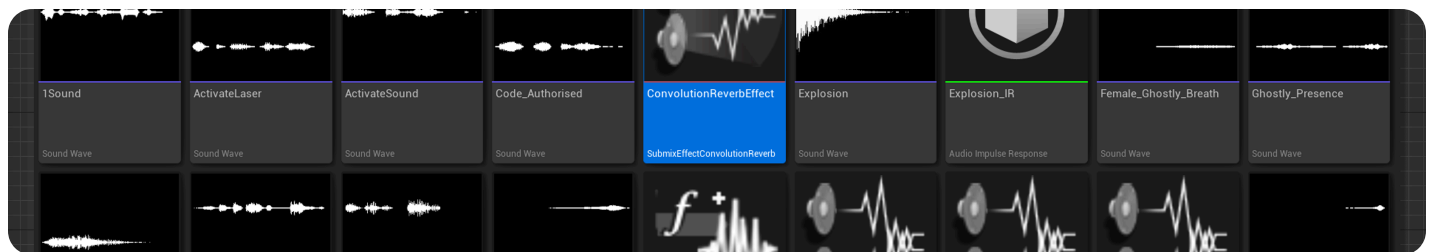


Developer
/ Documentation
/ Unreal Engine ▾
/ Unreal Engine 5.4 Documentation
/ Working with Audio
/ Reverb
/ Convolution Reverb

Convolution Reverb

An overview on using audio samples from physical spaces to simulate reverberation.



Convolution reverb is an exciting technology that sound designers can use to capture the **impulse responses (IRs)** of any physical location (such as caves, churches, offices, or hallways) and process audio in real time as though it were being played at that location. In effect, the IRs are *capturing* the reverberations of real locations, thereby allowing a new level of realism and immersion in environmental audio processing.

In addition to capturing real-life reverberations, IRs can be derived from any other sound source: algorithmic reverb output, foley, VO (such as huffs or breaths), contact mics, and so on. IRs can be edited and processed like any other sound — they can be filtered, attenuated, reversed, chopped-up, edited, and more. In this way, the Convolution Reverb feature introduces a whole new dimension of experimentation.

Convolution reverb provides a data-driven alternative to more traditional algorithmic reverb techniques that simulate reverberation using a combination of delay buffers, filters, and various other DSP topologies.

Conceptual Background

The Convolution Operation

Convolution is a mathematical operation like any other mathematical operation — addition, subtraction, integration, dot products, or cross products, to name a few. In the same way that dot products and cross products can be thought of as different ways to multiply vectors (or signals), convolution is a third way to think of vector transformation. As with cross products, convolving two signals results in a third signal.

Without getting overly involved in complex mathematics, the simplest way to conceptualize convolution is to consider every sample of the output signal, $y(t)$, as the sum of weights, $g(t)$, of the time-shifted input signal, $x(t)$. In this way, you can think of convolution as a kind of blurring operator.

Convolution in Audio

Convolution has a wide variety of applications in a large number of fields, including statistics, machine learning, and graphics rendering. In audio, convolution is a way to process audio signals through IRs, which represent real-world systems that would otherwise be impossible to model due to their complexity.

If the input audio signal is $f(t)$ and our weight, $g(t)$, is the IR that represents the complex system to be modeled, then the output, $x(t)$, is what the audio signal would be if it were fed through that system.

Computational Considerations

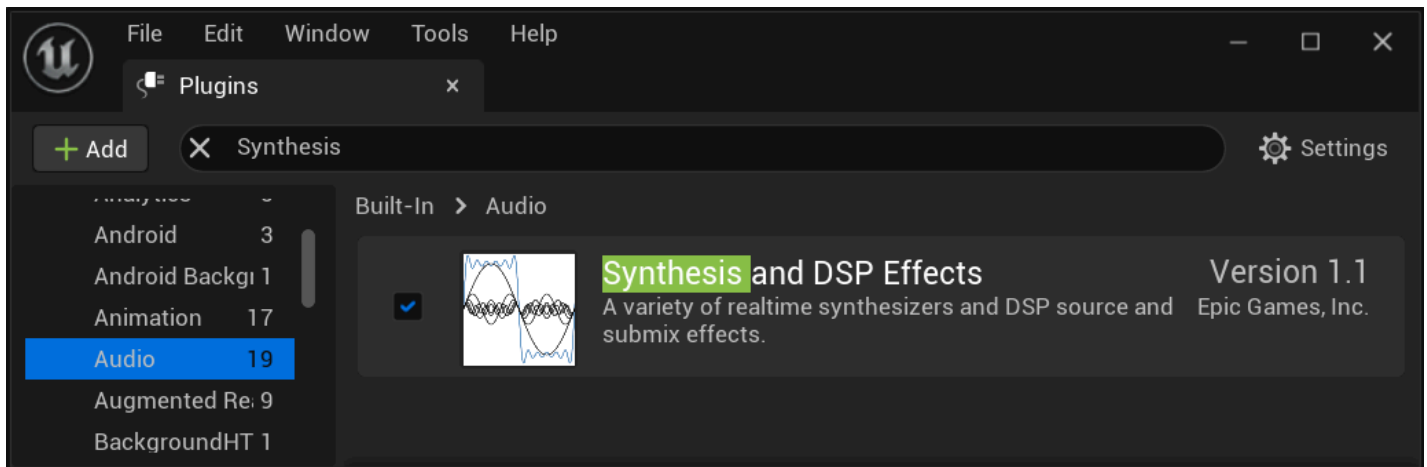
Convolution is fundamentally an expensive mathematical operation. Every element in the input signal, $f(t)$, is multiplied by every element in the convolving signal, $g(t)$, and added together to produce every element in the output signal, $x(t)$. For large IRs, which represent large reverberant spaces, this gets very expensive. Performing the convolution operation directly is usually too costly for a real-time scenario.

Luckily, there is a symmetry that can be taken advantage of that speeds up convolution significantly: convolving two signals in the *time domain* is identical to multiplying signals in the *frequency domain*. The frequency domain of a signal can be produced by using the **Fast Fourier Transform (FFT)**. The FFT, as the name would suggest, is fast. By doing this step for the input signal and the IR, convolution reverb can be computed on the CPU in real time. Take care, however, as it is still usually more expensive than traditional algorithmic reverbs.

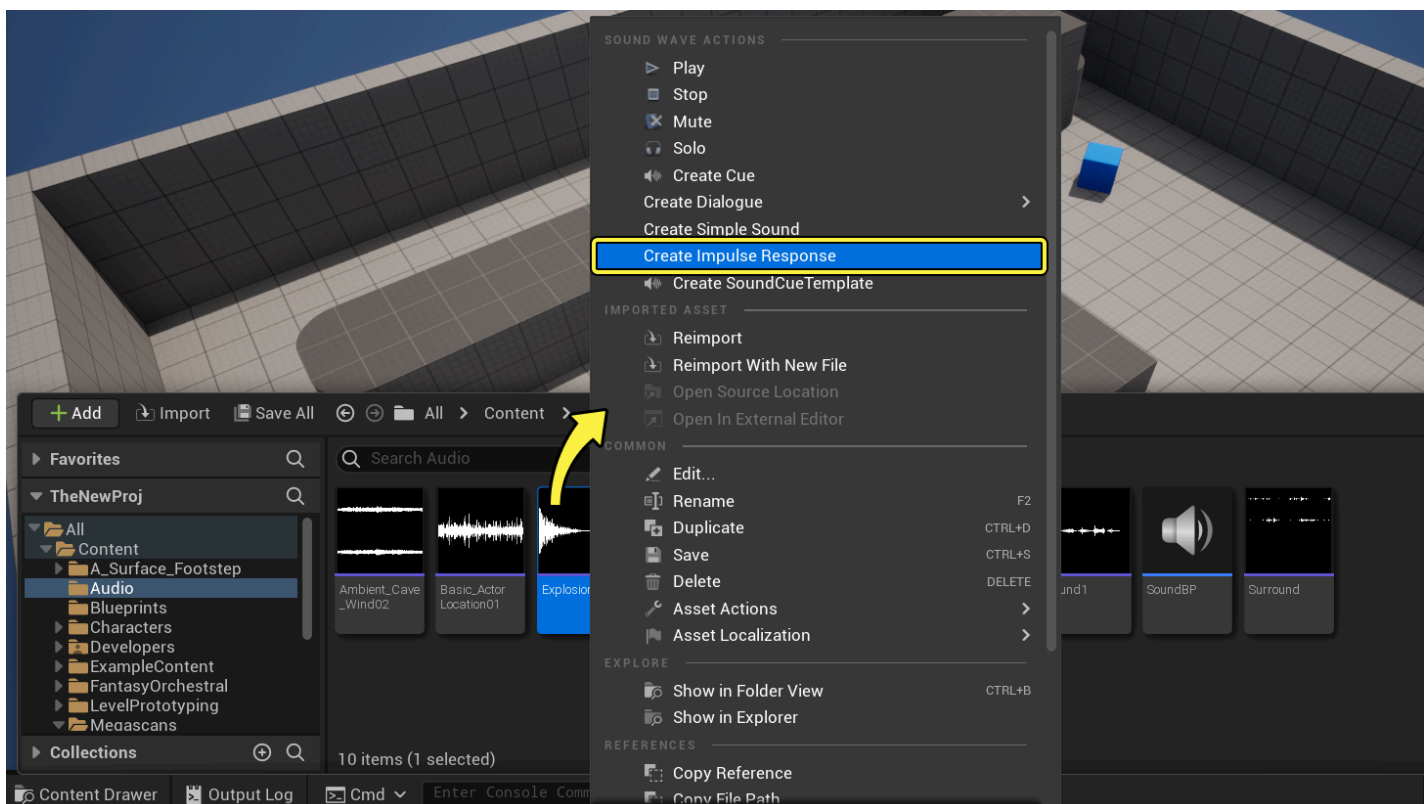
With new CPU capabilities and hardware acceleration of convolution operations, convolution reverb is becoming a more viable option in game audio. In this way, convolution reverb has been traditionally a next-gen audio technology.

How to Use the Convolution Reverb

Enable the **Synthesis and DSP Effects** plugin (**Edit > Plugins > Audio > Synthesis and DSP Effects**). This plugin contains a variety of optional synthesis and DSP capabilities.

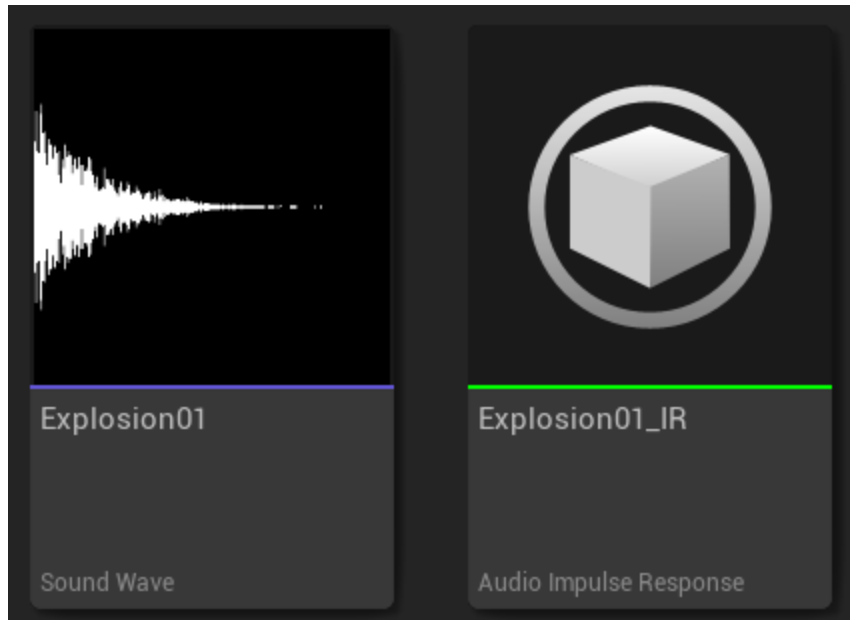


With the plugin enabled, right-click any imported sound wave in the Content Browser and select **Create Impulse Response**.



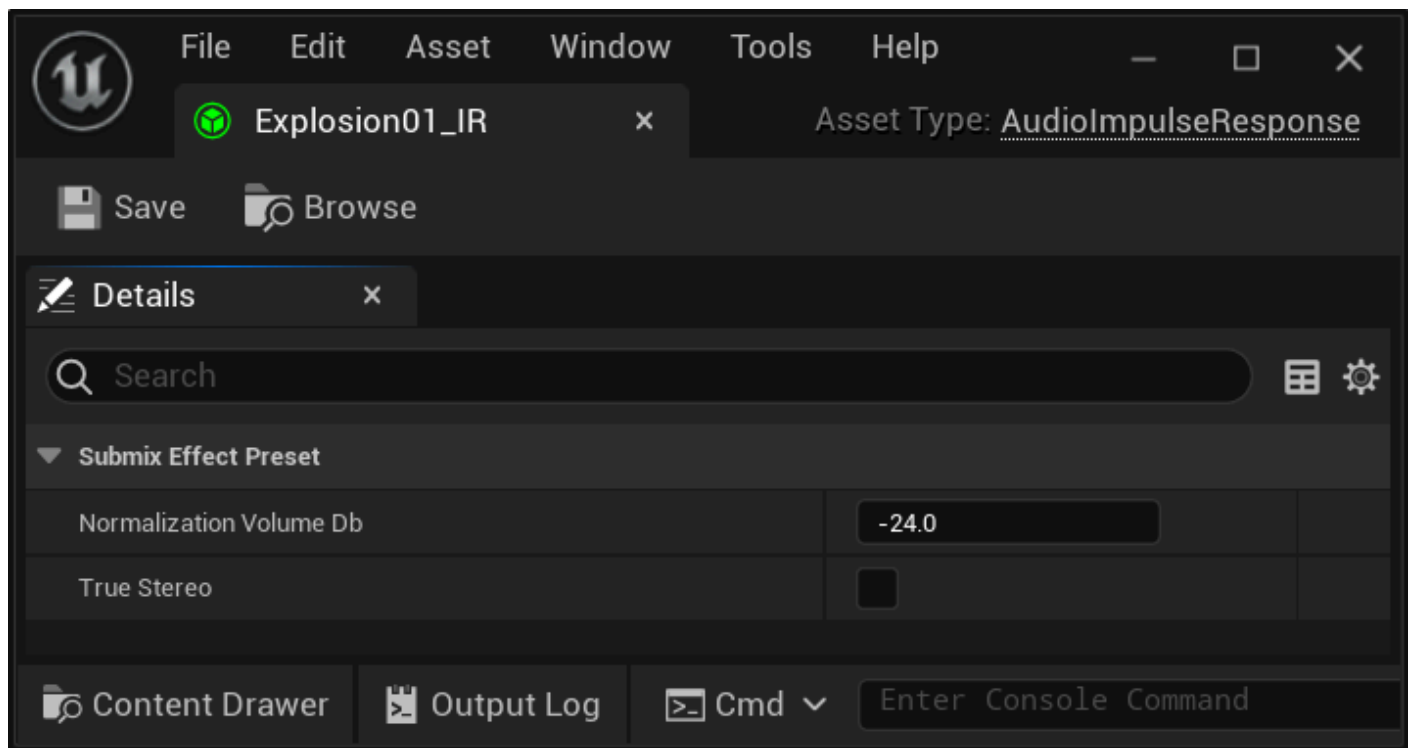
Click image for full size.

This creates an impulse response from the selected sound wave. In this way, any sound wave could become the IR for the convolution reverb algorithm (you can convolve with any sound). This will create a new asset with a name appended by **_IR**.



Import an IR with the same process you would use to import any sound wave.

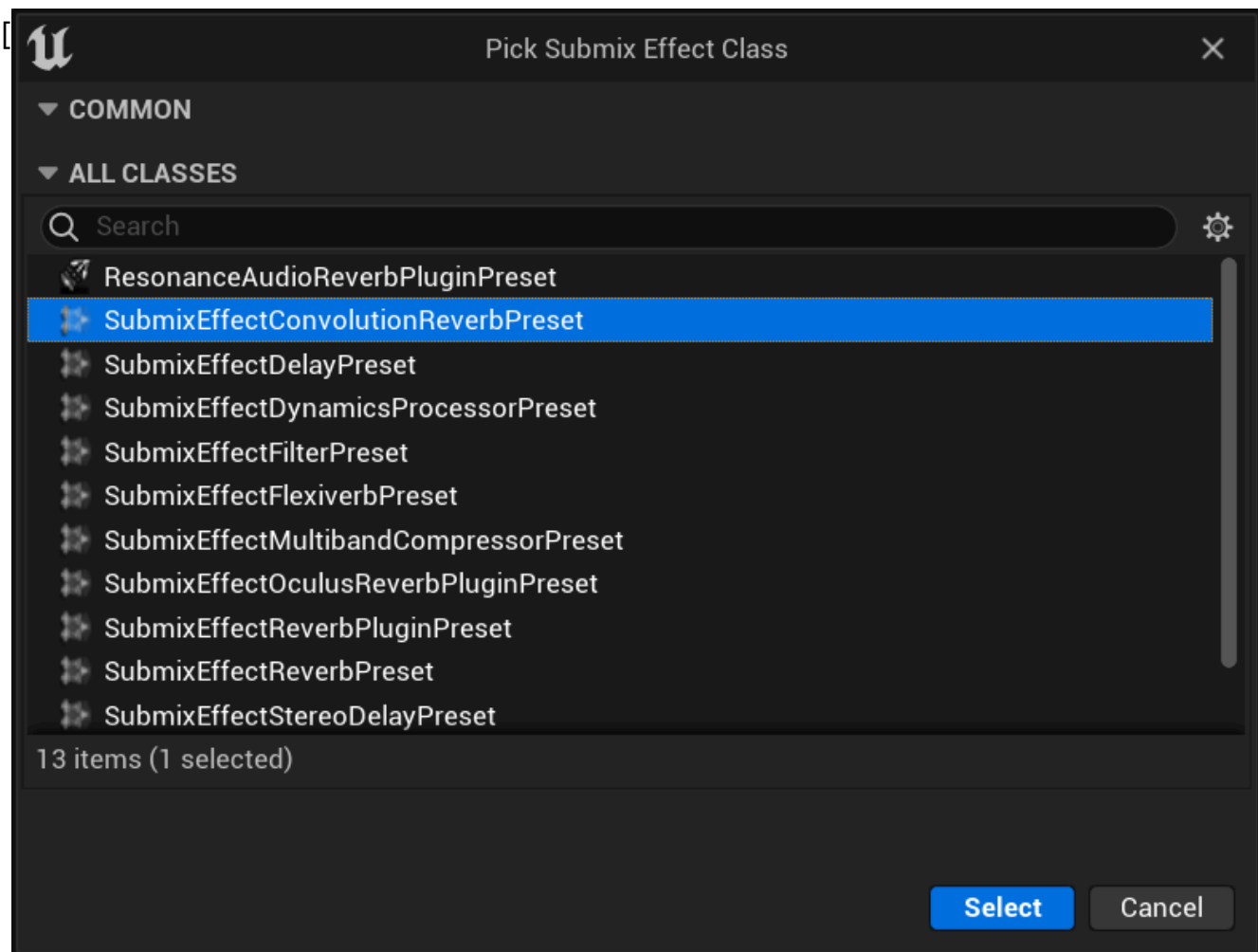
This asset contains some data that will be used by the convolution reverb effect, as well as a normalization value that can be set for each impulse response. To set the volume normalization value for an individual Audio Impulse Response asset, open the **Details** panel.



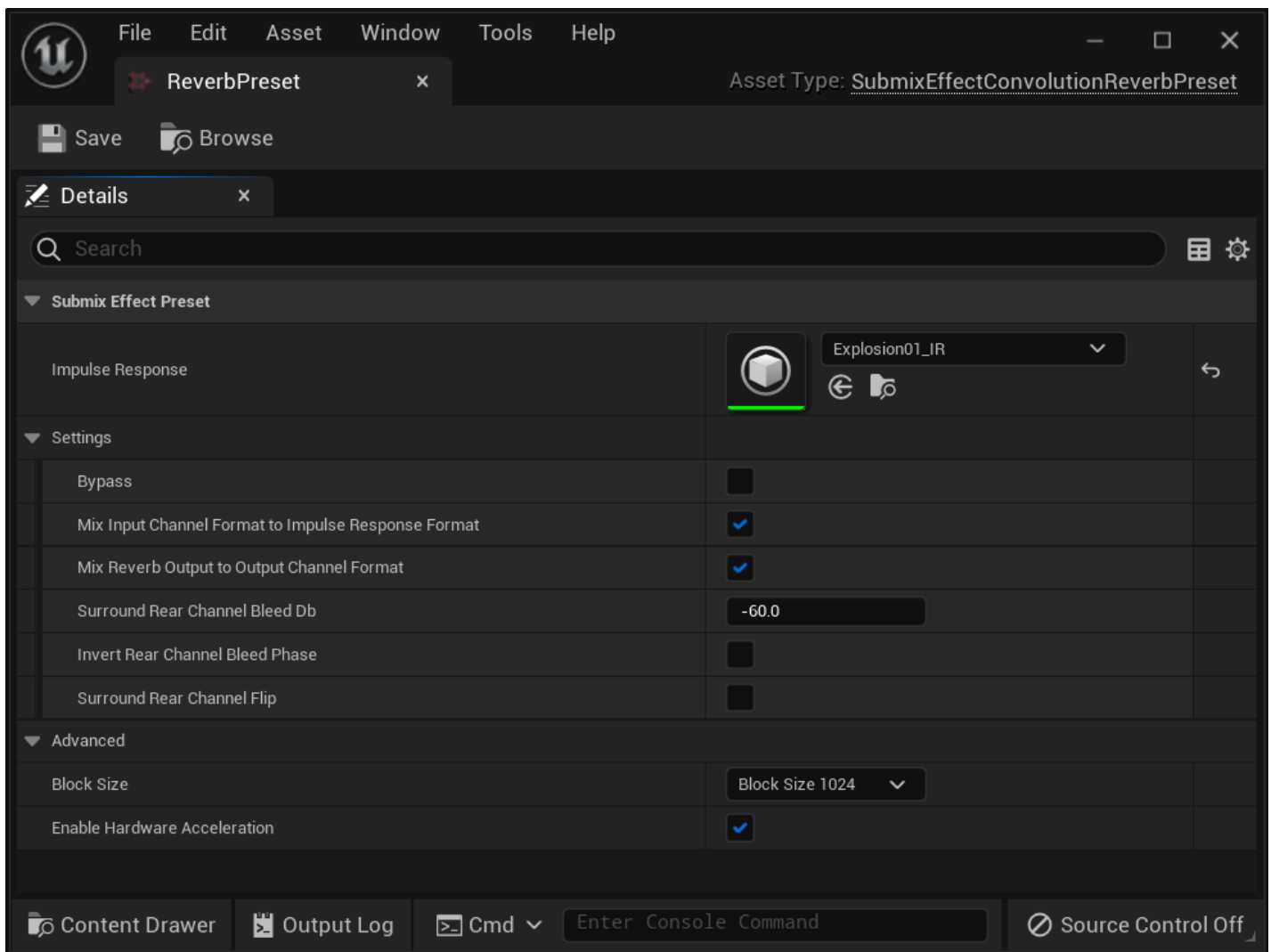
The **Normalization Volume Db** is a decibel attenuation that is applied to any convolution effect that is actively using this impulse response. This provides an easy way to achieve equal loudness between various impulse responses.

The **True Stereo** is a checkbox that defines whether impulse response channels are interpreted as true stereo or independent channel impulses.

To use the convolution reverb effect, create a **Submix Effect Preset** by right-clicking in the Content Browser and selecting **Audio > Effects > Submix Effect Preset**. When you create a new asset, the Pick Submix Class dialog will appear. Select the **SubmixEffectConvolutionReverbPreset**.



To open the **Details** panel for a Convolution Reverb Effect, double-click it in the Content Browser and set up the impulse response asset as the one to use.



Click image for full size.

Other parameters include:

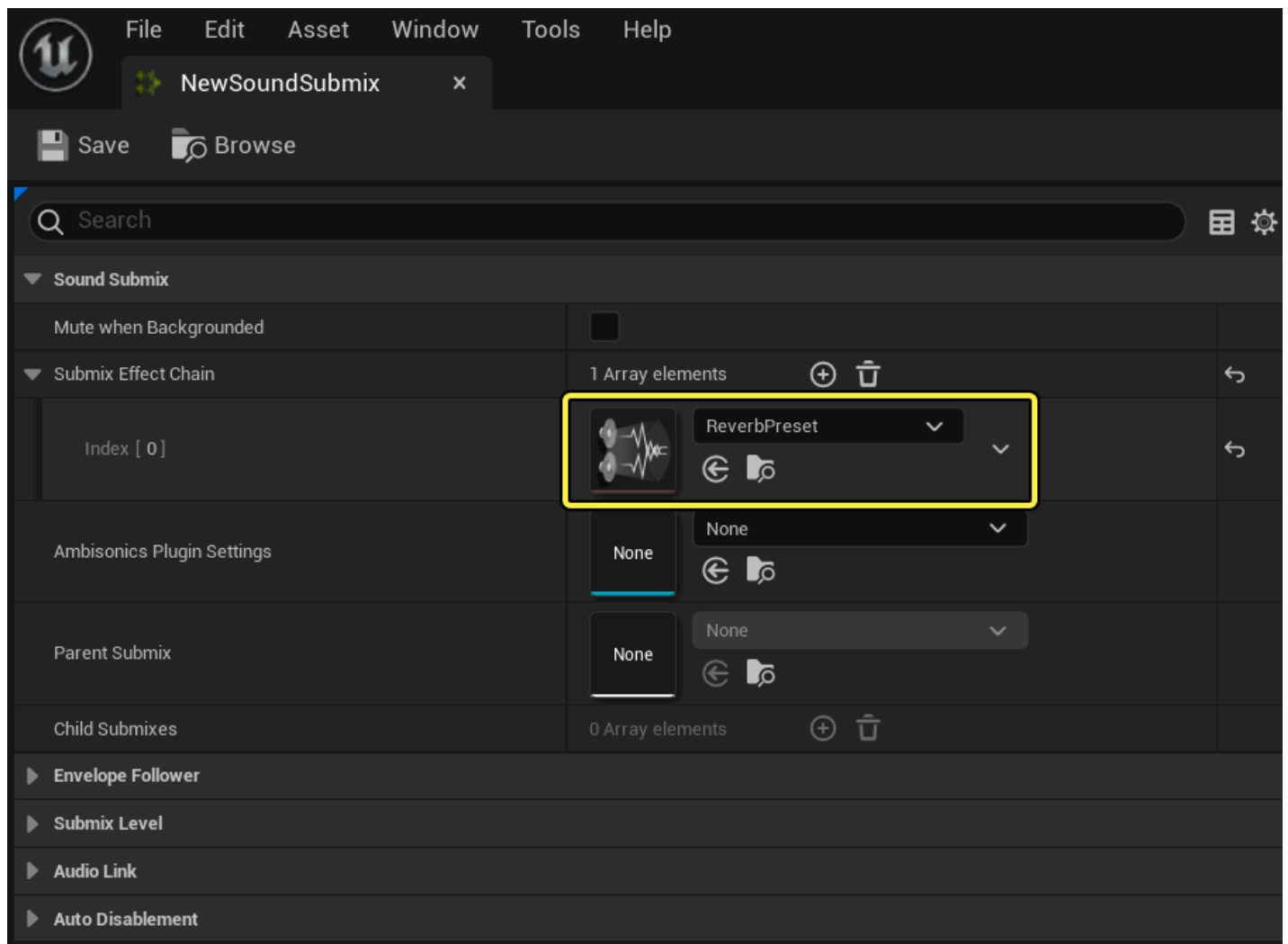
- **Bypass:** If true, the input audio is directly routed to the output audio with applying any effect.
- **Mix Input Channel Format to Impulse Response Format:** If true, the submix input audio is downmixed to match audio channel format of the impulse response asset. If false, the input audio channels are matched to the impulse response audio channels.
- **Mix Reverb Output to Output Channel Format:** If true, the reverberated audio is upmixed or downmixed to match the submix output audio format. If false, the reverberated audio channels are matched to the submix output audio channels.
- **Surround Rear Channel Bleed Db:** How many decibels of the output of the convolution effect to send to rear channels in a surround sound environment. Values of -60.0 or less will be considered silent.
- **Invert Rear Channel Bleed Phase:** Whether or not to invert the phase of the audio sent to the rear channels in surround sound environments.

- **Surround Rear Channel Flip:** Whether or not to flip the left and right channels of the effect before sending to the rear channels.

The advanced parameters include:

- **Block Size:** Sets the internal processing block size, which may have an effect on runtime performance.
- **Enable Hardware Acceleration:** Enables hardware acceleration if it exists on the target platform.

To use this on a submix effect, add it to any submix as a submix effect in a submix effect chain.



Click image for full size.

You can automatically send reverb audio to the submix if you set this submix as the master reverb submix to be used. Alternatively, you can manually send audio to this submix as a submix-send effect.