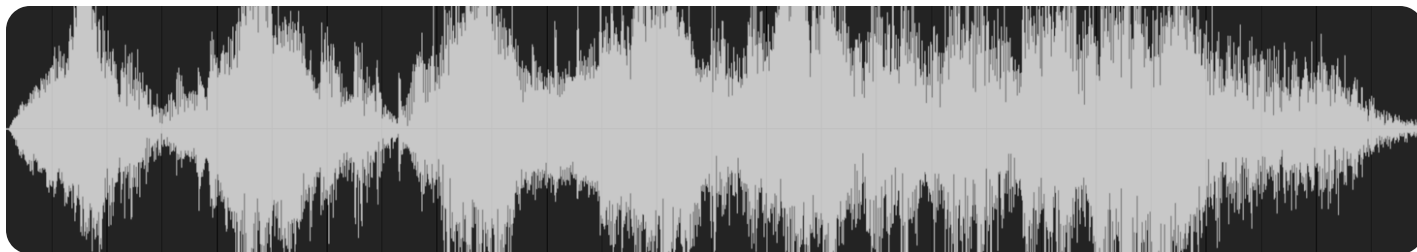


Importing Audio Files

Learn how to create Sound Wave assets by importing audio files.



Unreal Engine provides several features for you to create the desired audio for your project. **Sound Wave** assets are representations of audio files and are one of the basic building blocks required to work with many of those features. You can create Sound Waves by importing audio files into the Unreal Editor.

Supported Audio Files

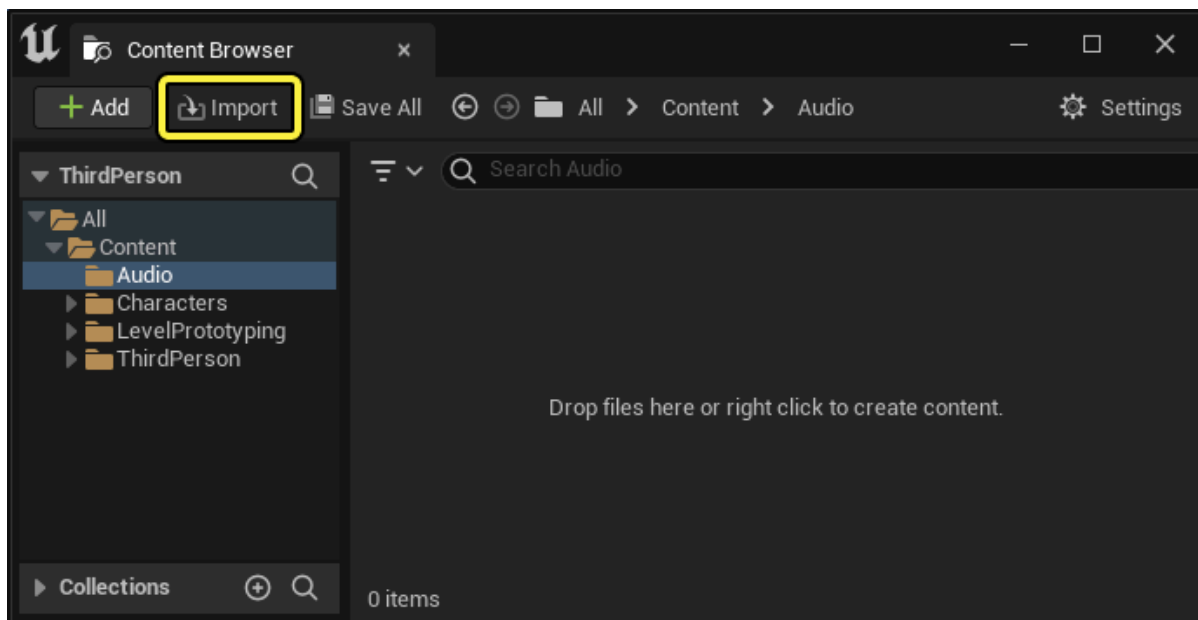
Format	<code>.wav</code> , <code>.ogg</code> , <code>.flac</code> , <code>.aif</code> , <code>.opus</code> , <code>.mp3</code>
Bit Depth	16, 24
Sample Rate	Any
Speaker Channels	Mono, Stereo, 4.0, 5.1, 7.1



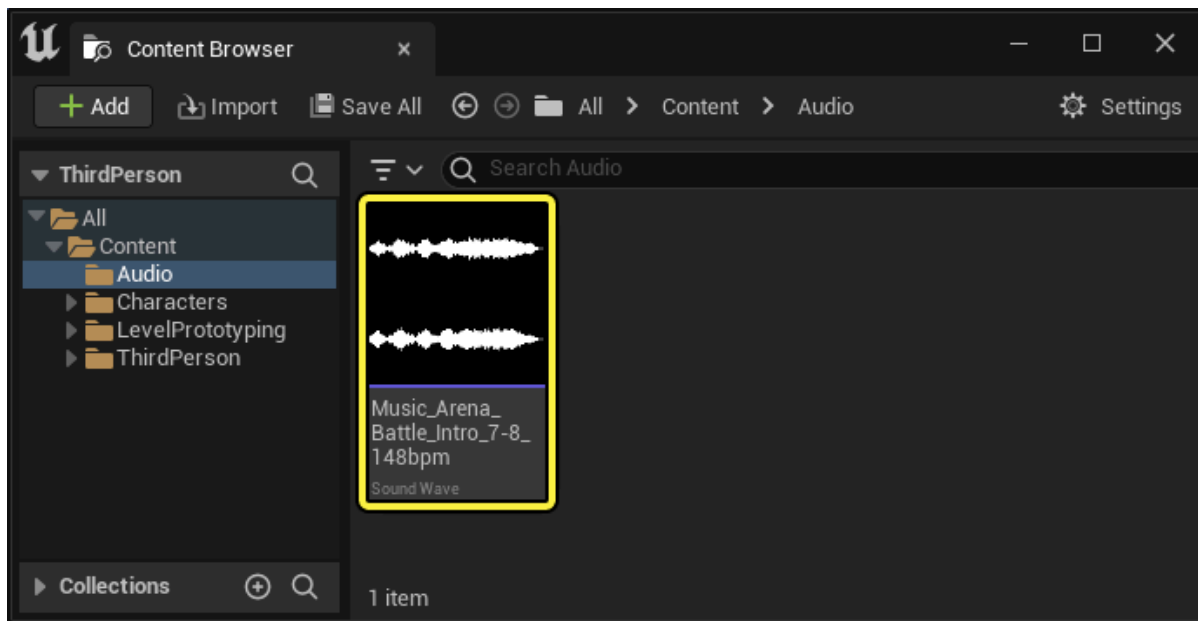
All imported audio files are converted to 16-bit `.wav` files internally. As such, exporting (by right-clicking a Sound Wave and selecting **Asset Actions > Export...**) will produce the converted file and not the originally imported file. Additionally, 24-bit files are not dithered during conversion so importing 16-bit files is recommended.

How to Import an Audio File

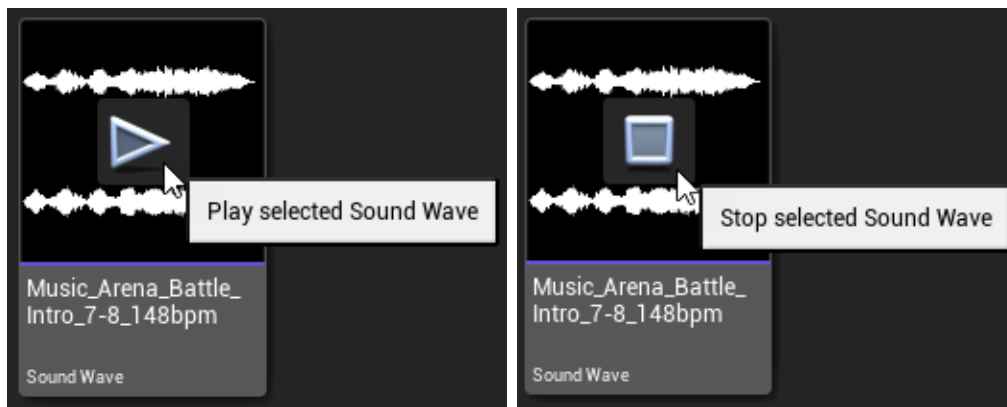
1. In the Content Browser, click the **Import** button.



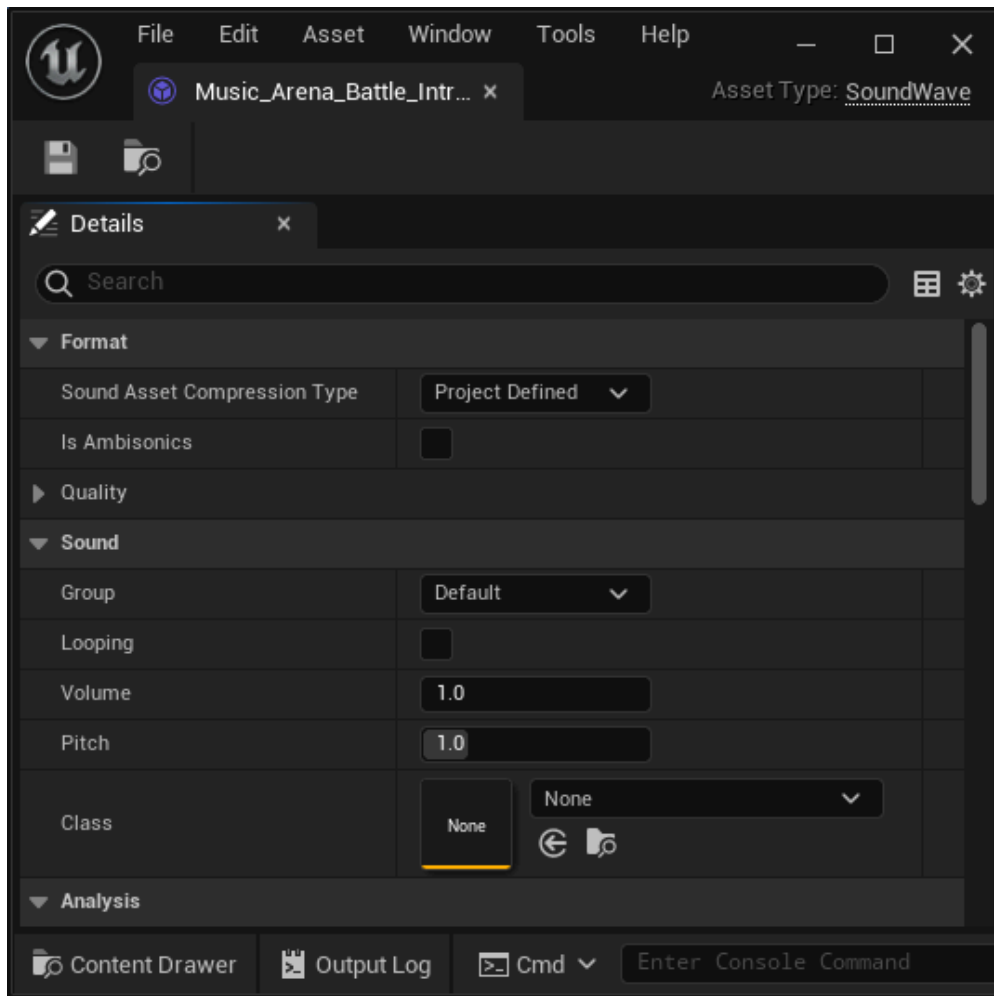
2. Select the file you wish to import using the **File Explorer**.
3. Find the newly created **Sound Wave** asset.



4. To preview the Sound Wave, hover over it until a **Play/Stop** toggle button appears and click it.



5. Double-click the Sound Wave to open the **Details** panel. You can view and edit the properties of your asset here.



You can also import audio files by dragging them from Windows Explorer directly into the **Content Browser**.



Unreal Engine also supports importing first-order ambisonic files. See [Native Soundfield Ambisonics Rendering](#) for information on importing and using ambisonic assets.

Compression

All Sound Waves are compressed with the **Default Audio Compression Type** specified in the **Project Settings**. You can also set the codec and quality level for each platform and Sound Wave.

Codec	Type	Ratio	Description
Bink Audio	Lossy	Up to 10:1	(Default) A perceptual-based codec that supports all features across all platforms. Low memory usage and comparable CPU usage to ADPCM .
ADPCM	Lossy	Up to 4:1	A time-domain codec with fixed-sized quality and is relatively cheap to decode.
PCM	Lossless	Uncompressed	Uses uncompressed audio that results in higher memory usage as streamed chunks contain less audio per chunk but provides extremely cheap decoding and support for all features.
Platform Specific	-	-	Encodes the asset in a platform-specific format. Seeking is not supported.