

AudioLink Overview

An overview of the AudioLink API.



⚠ Learn to use this **Beta** feature, but use caution when shipping with it.

AudioLink is an API that connects Unreal Audio Engine to external software, such as other audio engines. By using hardware abstraction, AudioLink bypasses the need for direct hardware access and provides the strengths of both Unreal Audio Engine and the connected software.

What Can AudioLink Transmit?

AudioLink transmits data through three link types.

- **Sources** - Outputs pulse-code modulation (PCM) data of pre-attenuated sources. Supported sources include MetaSounds, Sound Cues, and Sound Waves.
- **Submixes** - Outputs PCM data of a particular submix.
- **Audio Components** - Outputs PCM data from pre-attenuated sources used by an Audio Component. This is handled differently than standalone source and uses a different API.

How to Enable AudioLink

A custom setup is required based on how the implementer provides functionality. Typically, the implementer registers an **AudioLink Factory** during the Unreal Engine (UE) start-up phase, which enables AudioLink.

In Unreal Editor, each link type has a **Send to AudioLink** flag to enable transmission and an **AudioLink Settings** parameter to optionally override default settings. These options persist even if AudioLink is disabled, so nothing is lost.

Enabling Source Transmission

The AudioLink properties are accessible through the **Sound Attenuation** asset associated with the source. In the source's **Details** panel, find or create the associated asset under **Attenuation > Attenuation Settings**.

You can set the **Enable Send to AudioLink** flag and the **AudioLink Settings Override** property in the Sound Attenuation asset's **Details** panel, under **Attenuation (AudioLink)**.



Despite the name, the Sound Attenuation asset functions as a collection of source properties and is not for attenuation exclusively.

Enabling Submix Transmission

You can set the **Send to Audio Link** flag and the **Audio Link Settings** property in the submix's **Details** panel, under **Audio Link**.



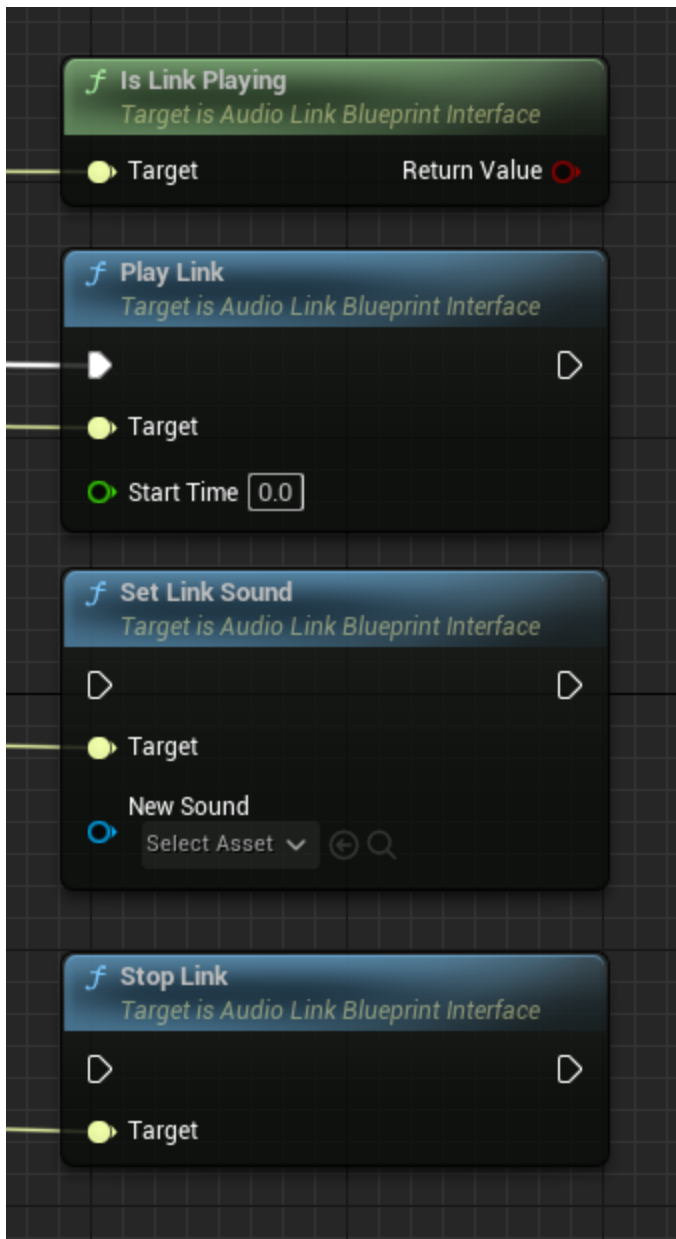
Combining the same submixes and sources can cause stacking and loud results in the mixed audio.

Enabling Audio Component Transmission

Audio Component support is included with AudioLink, but is often unnecessary as sources are componentless by design, highly optimized for scale, and provide engine-level support. However, sometimes it's necessary to use Audio Components, such as for raycasting spatialization.

Audio Component transmission requires a custom approach depending on the implementation. However, the Blueprint interface is standardized. Implement

`AudioLinkBlueprintInterface` to provide access to the following API.





- On the **SetLinkSound** function, you can set **NewSound** to any `USoundBase` object, including a MetaSound, Sound Wave, or Sound Cue.
- We deliberately named the **PlayLink** function so you can include it in Components with other "Play" calls. If necessary, you can redirect the PlayLink function.

How to Disable AudioLink

You can disable AudioLink with the `au.audiolink.enabled` console variable.

Limitations

- Running more than one audio engine comes at an additional overhead cost. Although Unreal Audio Engine is highly optimized, we recommend that you evaluate usage in your project.
- Due to hardware sharing limitations, hardware decoders are not supported for sources running inside UE. However, UE supports efficient software decoders, like Bink Audio and Ogg Vorbis.
- AudioLink currently only supports sources with up to eight channels.
- Ambisonic sources or object-based formats are not supported at this time.

Troubleshooting

- Use the console command `log LogAudioLink All` to output more detailed information to the log.
- Make sure you register the AudioLink factory with UE. Log messages will indicate if this is working correctly.
- Ensure you enable the **Send to AudioLink** flag on the source or submix.
- Be careful with pairing a source and submix containing the same signal chain, as that can cause data duplication and result in loud and undesirable audio stacking.

- Each editor and game window, including Play In Editor (PIE), has its own Unreal Audio Engine instance. This is an extremely powerful concept but can cause confusion about instance lifetimes. For example, the editor's submixes are instantiated on startup. This happens early on but an AudioLink Factory can miss a submix if it is not instantiated early in the flow. Fine-tune your implementation accordingly.