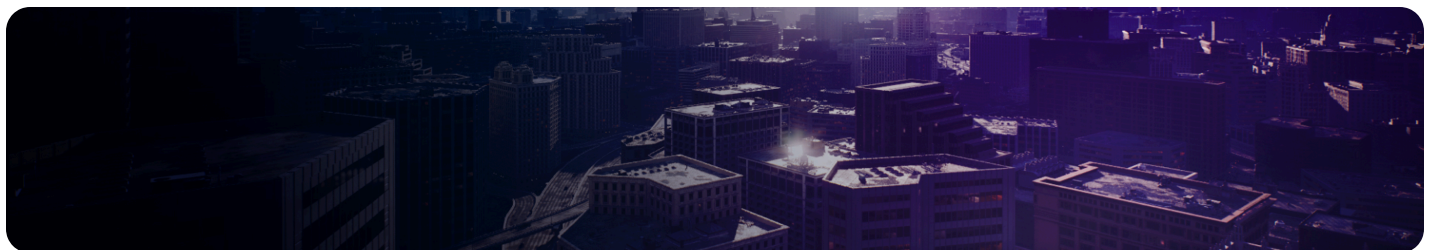


- Developer
- / Documentation
- / Unreal Engine ▾
- / Unreal Engine 5.4 Documentation
- / Working with Audio
- / Audio Memory Management
- / Sound Concurrency Reference Guide

Sound Concurrency Reference Guide

A reference guide for Sound Concurrency assets.



Playing too many sounds at the same time can result in high resource consumption and poor performance. **Sound Concurrency** assets are one of the primary tools you can use to handle that problem. With a Sound Concurrency asset, you can limit how many sounds play simultaneously and what to do when that limit is reached. You can assign these rules to individual sounds, groups of sounds, or all sounds within a project.

Creating Sound Concurrency Assets

You can create a Sound Concurrency by clicking the **Add** button in the **Content Browser** and selecting **Audio > Sound Concurrency**.

Editing Sound Concurrency Assets

The screenshot shows a 'Details' panel with a search bar and a settings icon. It is divided into three sections: 'Concurrency', 'Volume Scaling', and 'Voice Stealing'. Each section contains several properties with input fields or checkboxes.

Concurrency	
Max Count	16
Limit to Owner	<input type="checkbox"/>
Resolution Rule	Stop Farthest Then Oldest ▼
Retrigger Time	0.0

Volume Scaling	
Can Recover	<input type="checkbox"/>
Volume Scale	1.0
Volume Scale Mode	Default ▼
Duck Time	0.01
Recover Time	0.5

Voice Stealing	
Voice Steal Release Time	0.0

You can edit the properties of the Sound Concurrency by double-clicking it in the **Content Browser**, or by right-clicking it and selecting **Edit...** from the context menu. A **Details** panel for the asset will appear.

Details

Concurrency

Property	Description
Max Count	The maximum number of concurrently active sounds allowed in this group before the Resolution Rule takes effect.
Limit to Owner	If enabled, limits these concurrency rules to each actor that plays the sound.
Resolution Rule	Sets the behavior to follow when Max Count has been reached: <ul style="list-style-type: none"> • Prevent New: Prevent any new sounds from starting. • Stop Oldest: Stop the oldest playing sound. • Stop Farthest Then Prevent New: Stop the farthest sound. If all sounds are the same distance, prevent new sounds from starting.

Property	Description
	<ul style="list-style-type: none"> • Stop Farthest Then Oldest: Stop the farthest sound. If all sounds are the same distance, stop the oldest sound instead. • Stop Lowest Priority: Stop the lowest priority sound. If all sounds have the same priority, stop the oldest sound instead. • Stop Quietest: Stop the quietest sound. • Stop Lowest Priority Then Prevent New: Stop the lowest priority sound. If all sounds have the same priority, prevent new sounds from starting.
Retrigger Time	The wait time (in seconds) between playing sounds. The sounds rejected by this setting ignore virtualization settings.



All active components count towards **Max Count**, not just actively playing sounds. This can affect systems that can remain active without playing any audio, such as synthesizers. It can also affect **Source Buses** as they take up two spots within your **Max Count**: one for the original source and one for the bus. This is true even if the source is set to **Output to Bus Only** because it is still an active component.

Volume Scaling

Property	Description
Can Recover	If enabled, allows volume scaling to recover to its default value after all sounds in the group have stopped playing.
Volume Scale	The scaling factor (ducking) applied to older sounds, which compounds based on the Volume Scale Mode .

Property	Description
Volume Scale Mode	Sets the volume scaling behavior to use based on the attributes of the active sounds in the group: <ul style="list-style-type: none"> • Default: Scale older sounds more than newer sounds. • Distance: Scale distant sounds more than closer sounds. • Priority: Scale lower priority sounds more than closer sounds.
Duck Time	The time (in seconds) to duck using the Volume Scale .
Recover Time	The time (in seconds) to recover from volume scaling.

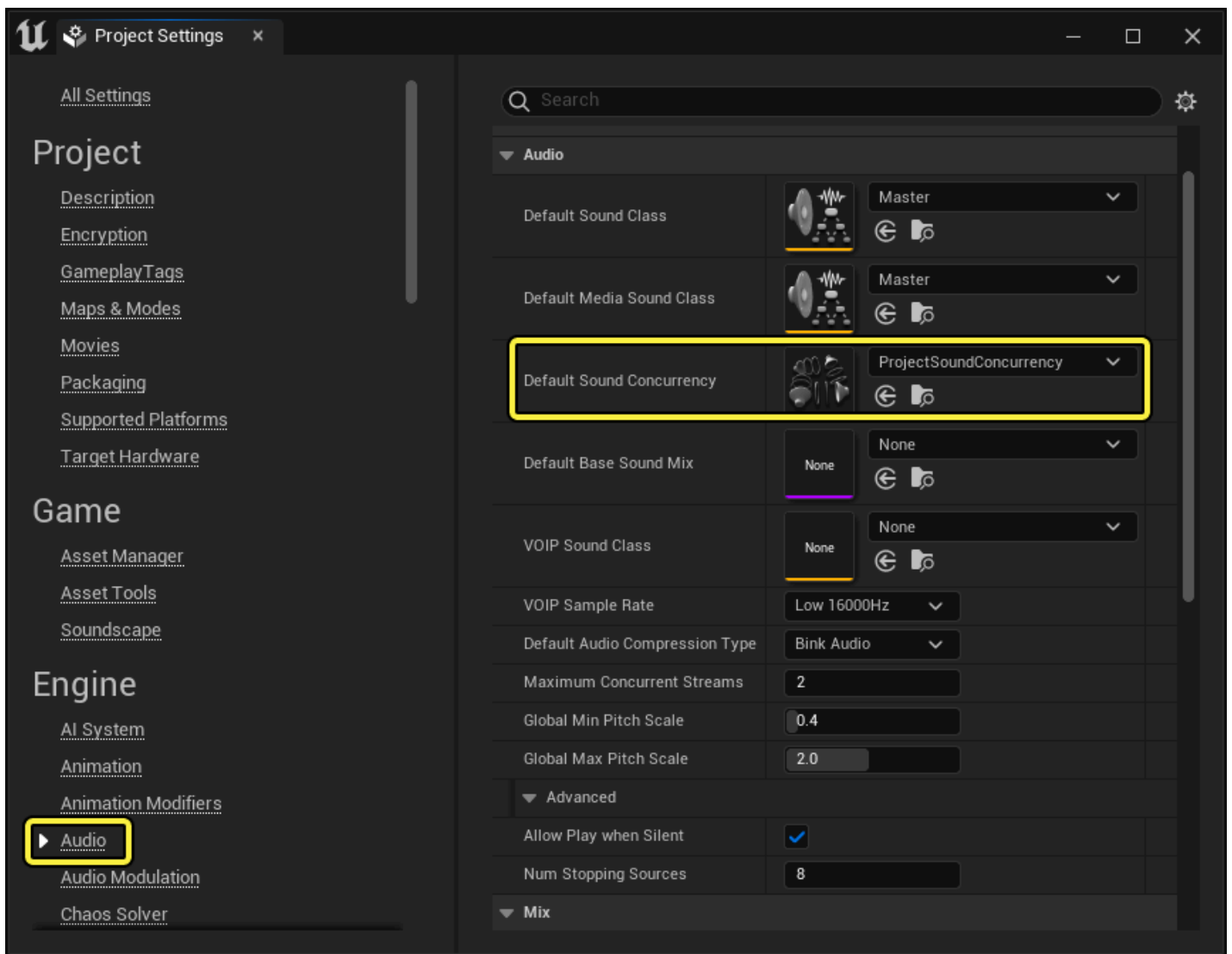


To avoid thrashing sounds, **Volume Scale** is applied after **Stop Quietest** rules are evaluated.

Voice Stealing

Property	Description
Voice Steal Release Time	The fade-out time (in seconds) used if a sound is stopped due to another sound in the group starting.

Setting Sound Concurrency Assets In Project Settings

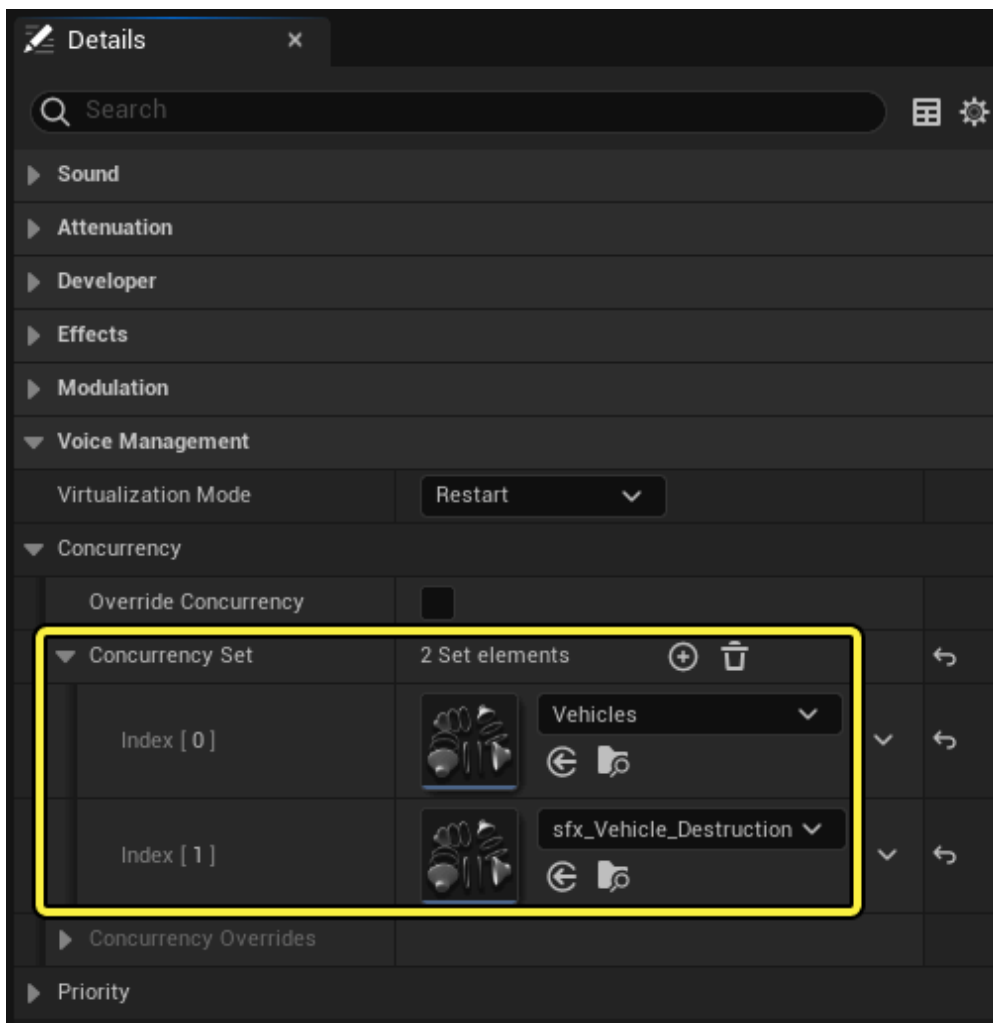


You can set a Sound Concurrency as the default for all sound sources in the **Project Settings**.

To assign a project default Sound Concurrency:

1. Go to **Edit > Project Settings...** to open the **Project Settings** panel.
2. On the left side of the panel, click **Audio** under the Engine heading.
3. Click the **Default Sound Concurrency** dropdown and select the Sound Concurrency to use.

On Source Assets



You can also set Sound Concurrency assets directly on sound source assets, such as **MetaSounds**, **Sound Cues**, and **Sound Waves**.

To assign a Sound Concurrency to a sound source:

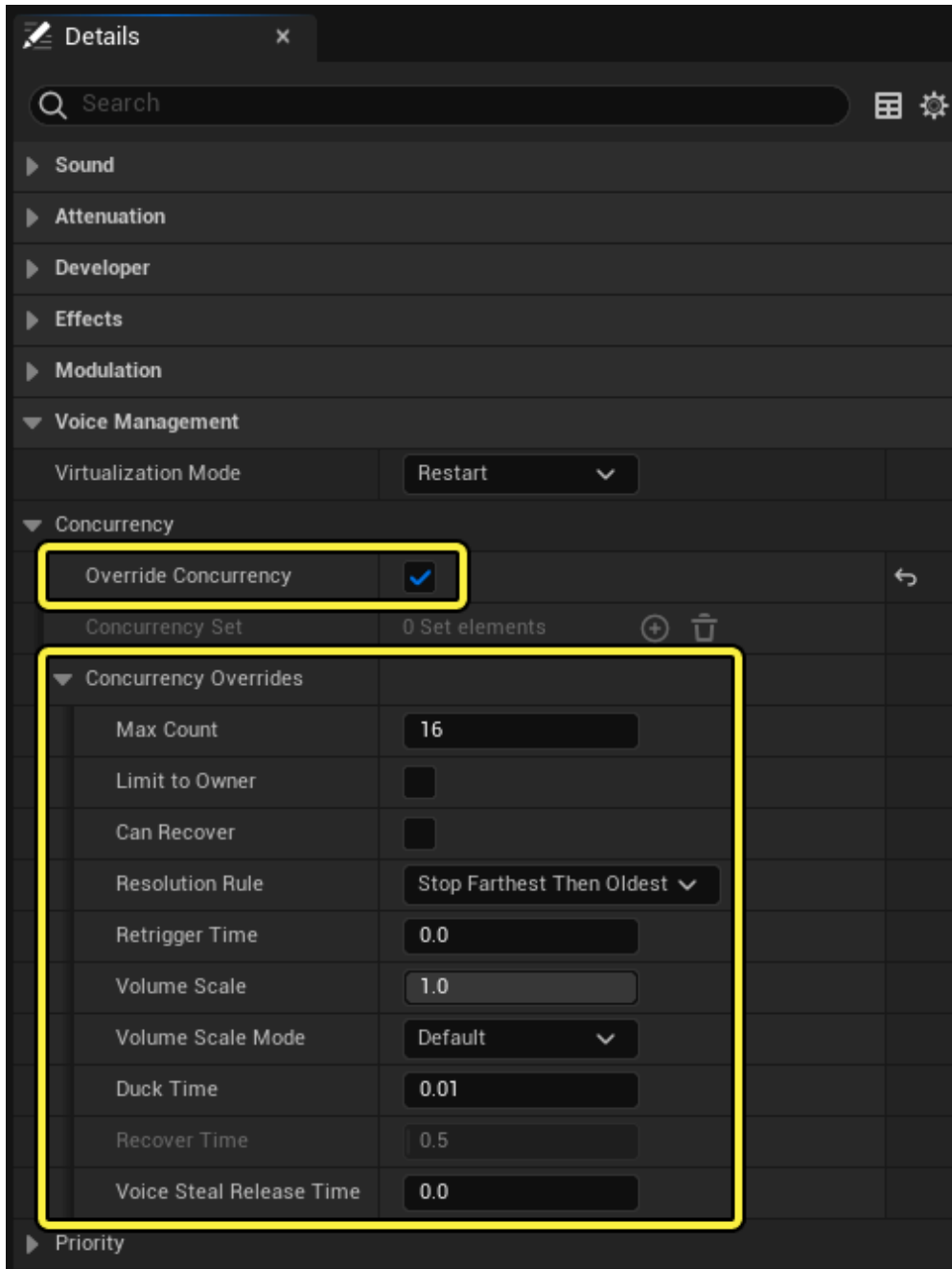
1. Open the sound source's **Details** panel.
2. Find **Voice Management > Concurrency > Concurrency Set** in the **Details** panel
3. Click the **Add** button to add an index to the **Concurrency Set**.
4. Click the dropdown for the new index and select the Sound Concurrency to use.

You can use the same Sound Concurrency across multiple sound source assets. If you do so, the properties and state of the Sound Concurrency are shared between the sources.

You can specify multiple Sound Concurrency assets within a **Concurrency Set**. The source must fulfill all of the rules in the set before it can play.



If a **Concurrency Set** has multiple Sound Concurrency assets with fulfilled rules that stop active sounds, one sound per group will be stopped to make room for the new sound.



You can also enable the **Override Concurrency** option and use the properties under **Concurrency Overrides** to specify behavior for an individual sound asset without creating a Sound Concurrency asset.