

FoxSound

Music, Crossfade & Region Audio Toolkit for Blueprints

User Guide

Version 1.0

Alchemy Fox Studio

Supported Engine: UE 5.4, 5.6, 5.7, 5.8

Table of Contents

■ Installation

■ Quick Start Tutorial

■ Node Reference

1. Region (10 Nodes)
2. Music (6 Nodes)
3. Crossfade (5 Nodes)
4. Spatial (4 Nodes)
5. Utility (7 Nodes)
6. Volume (4 Nodes)

■ Implementation Notes

Installation

Option A: Via Unreal Engine Tab

1. Open the Epic Games Launcher and go to the Unreal Engine tab.
2. Search for FoxSound.
3. Click **Install to Engine**, select your Engine version, and click **Install**.
4. Open your project, go to **Edit > Plugins**, enable FoxSound, then restart the editor.

Option B: Via Fab Library

1. Open the Epic Games Launcher and go to **Fab**.
2. Open **My Library**, search for FoxSound and click on it.
3. Click **Install Plugin**, select your Engine version, and click **Install**.
4. Open your project, go to **Edit > Plugins**, enable FoxSound, then restart the editor.

Option C: Manual Installation

1. Copy the FoxSound/ folder into **YourProject/Plugins/**.
2. Right-click the .uproject file and select **Generate Visual Studio project files**.
3. Open the project, go to **Edit > Plugins**, enable FoxSound, then restart the editor.

All nodes appear under **FoxSound | <Category>** when you right-click in any Blueprint graph.

Supported Engine Version: **UE 5.4, 5.6, 5.7, 5.8**

Quick Start Tutorial

Every FoxSound node is a static Blueprint function — no setup, no initialization, no subsystem required (unless noted). Below are step-by-step examples showing how to use FoxSound in real project scenarios.

Example 1: Dynamic Music Regions in an Open World

Goal: Automatically crossfade between calm forest music and tense dungeon music as the player moves between areas.

Register Sound Regions

In your GameMode BeginPlay, call Register Sound Region twice: Name="Forest" with your forest music + ambient wind track, and Name="Dungeon" with dungeon music + drip ambient track.

Enter/Exit on Trigger Overlap

On your Forest TriggerVolume's BeginOverlap, call Enter Sound Region ("Forest", CrossfadeDuration=3.0). On EndOverlap, call Exit Sound Region.

Stack Sub-Regions

Inside the dungeon, use Push Region Stack for a "DungeonBoss" sub-region. Exiting pops back to the regular dungeon music automatically — no manual track swapping.

Play and Test

Walk between areas — music crossfades smoothly over 3 seconds. Enter the boss room — boss music fades in. Leave — regular dungeon music returns. Zero AudioComponent management.

Example 2: Combat Music on Enemy Detection

Goal: Transition to a fast combat track when the player is spotted, then fade back to exploration music when safe.

Register Combat Track

When registering your "Forest" region, also set CombatTrack to your battle music asset. The region handles both peaceful and combat states on the same region entry.

Trigger Combat State

In your enemy AI's "OnPlayerSpotted" event, call Set Combat State with bInCombat=true, CrossfadeDuration=1.5. Music crossfades in 1.5 s.

Return to Peaceful

When all enemies lose the player (or die), call Set Combat State with bInCombat=false. The crossfade back to peaceful music begins automatically.

Add Danger State

For "enemies nearby but no combat yet", use Set Danger State (true/false) with your DangerTrack. This creates a 3-tier music system: Peaceful → Danger → Combat.

Example 3: Randomized Footstep Sounds

Goal: Play a random footstep from a pool at a character's foot bone location on each step.

Set Up a Notified Anim

In your walk/run AnimMontage, add AnimNotify events at heel-strike frames. Name them "FootstepLeft" and "FootstepRight".

Handle the Notify in Blueprint

In your Character's AnimBP, handle the notify. Get the foot bone world location using FoxAnim's Get Bone World Transform (BoneName="foot_l" or "foot_r").

Play Random Sound

At the bone location, call Play Random Sound at Location with your Sounds array (4–6 footstep variations), Volume=0.8, PitchVariation=0.1. Each step uses a random sound with slight pitch variation.

Play and Test

Walk around — footsteps are randomized and positioned at the actual foot location. PitchVariation prevents the robotic repetition of a single sound.

Node Reference

Complete reference for all 36 FoxSound nodes organized by category. Each node includes a description, inputs, outputs, and a use case hint.

1. Region (10 Nodes)

1.1 Register Sound Region

Provides 1.1 Register Sound Region functionality.

Input	Type	Description
WorldContext	Object Reference	
RegionName	Name	
MusicTrack	Sound Base	
AmbientTrack	Sound Base	
CombatTrack	Sound Base	
DangerTrack	Sound Base	

(none)

Use Case: Region helper — manage sound regions and track switching.

1.2 Enter Sound Region

Provides 1.2 Enter Sound Region functionality.

Input	Type	Description
WorldContext	Object Reference	
RegionName	Name	
CrossfadeDuration	Float	Time in seconds

(none)

Use Case: Region helper — manage sound regions and track switching.

1.3 Exit Sound Region

Provides 1.3 Exit Sound Region functionality.

Input	Type	Description
WorldContext	Object Reference	
CrossfadeDuration	Float	Time in seconds

(none)

Use Case: Region helper — manage sound regions and track switching.

1.4 Set Combat State

Provides 1.4 Set Combat State functionality.

Input	Type	Description
WorldContext	Object Reference	
bInCombat	Bool	Boolean flag
CrossfadeDuration	Float	Time in seconds

(none)

Use Case: Region helper — manage sound regions and track switching.

1.5 Set Danger State

Provides 1.5 Set Danger State functionality.

Input	Type	Description
WorldContext	Object Reference	
bInDanger	Bool	Boolean flag
CrossfadeDuration	Float	Time in seconds

(none)

Use Case: Region helper — manage sound regions and track switching.

1.6 Get Current Region

Provides 1.6 Get Current Region functionality.

Input	Type	Description
WorldContext	Object Reference	

Output	Type	Description
RegionName	Name	
State	EFoxSoundState	
MusicTrack	Sound Base	
AmbientTrack	Sound Base	

Use Case: Region helper — manage sound regions and track switching.

1.7 Set Region Ambient Volume

Provides 1.7 Set Region Ambient Volume functionality.

Input	Type	Description
WorldContext	Object Reference	
Volume	Float	

(none)

Use Case: Region helper — manage sound regions and track switching.

1.8 Override Region Track

Provides 1.8 Override Region Track functionality.

Input	Type	Description
WorldContext	Object Reference	
TrackType	EFoxRegionTrackType	
NewTrack	Sound Base	
CrossfadeDuration	Float	Time in seconds

(none)

Use Case: Region helper — manage sound regions and track switching.

1.9 Clear Region Override

Provides 1.9 Clear Region Override functionality.

Input	Type	Description
WorldContext	Object Reference	
TrackType	EFoxRegionTrackType	
CrossfadeDuration	Float	Time in seconds

(none)

Use Case: Region helper — manage sound regions and track switching.

1.10 Push Region Stack

Provides 1.10 Push Region Stack functionality.

Input	Type	Description
WorldContext	Object Reference	
RegionName	Name	
CrossfadeDuration	Float	Time in seconds

(none)

Use Case: Region helper — manage sound regions and track switching.

2. Music (6 Nodes)

2.1 Play Music

Provides 2.1 Play Music functionality.

Input	Type	Description
WorldContext	Object Reference	
Sound	Sound Base	
FadeInDuration	Float	Time in seconds
Volume	Float	

(none)

Use Case: Music control — play, switch, pause or query the music track.

2.2 Switch Music

Provides 2.2 Switch Music functionality.

Input	Type	Description
WorldContext	Object Reference	
NewSound	Sound Base	
CrossfadeDuration	Float	Time in seconds
Volume	Float	

(none)

Use Case: Music control — play, switch, pause or query the music track.

2.3 Stop Music

Provides 2.3 Stop Music functionality.

Input	Type	Description
WorldContext	Object Reference	
FadeOutDuration	Float	Time in seconds

(none)

Use Case: Music control — play, switch, pause or query the music track.

2.4 Pause Resume Music

Provides 2.4 Pause Resume Music functionality.

Input	Type	Description
WorldContext	Object Reference	
bPause	Bool	Boolean flag
FadeDuration	Float	Time in seconds

(none)

Use Case: Music control — play, switch, pause or query the music track.

2.5 Set Music Volume

Provides 2.5 Set Music Volume functionality.

Input	Type	Description
WorldContext	Object Reference	
Volume	Float	

(none)

Use Case: Music control — play, switch, pause or query the music track.

2.6 Is Music Playing

Provides 2.6 Is Music Playing functionality.

Input	Type	Description
WorldContext	Object Reference	
Output	Type	Description
Return Value	Bool	
CurrentTrack	Sound Base	

Use Case: Music control — play, switch, pause or query the music track.

3. Crossfade (5 Nodes)

3.1 Crossfade Sounds

Provides 3.1 Crossfade Sounds functionality.

Input	Type	Description
WorldContext	Object Reference	
FadeOutComponent	UAudioComponent*	Component reference
FadeInComponent	UAudioComponent*	Component reference
Duration	Float	Time in seconds
Output	Type	Description
Return Value	UFoxCrossfadeSoundsAction*	

Use Case: Crossfade action — smooth transition between sound assets.

3.2 Stop Sound With Fade Out

Provides 3.2 Stop Sound With Fade Out functionality.

Input	Type	Description
AudioComponent	UAudioComponent*	Component reference
FadeOutDuration	Float	Time in seconds
Output	Type	Description
Return Value	UFoxStopSoundFadeAction*	

Use Case: Crossfade action — smooth transition between sound assets.

3.3 Transition Sound

Provides 3.3 Transition Sound functionality.

Input	Type	Description
WorldContext	Object Reference	
OldComponent	UAudioComponent*	Component reference
NewSound	Sound Base	
CrossfadeOverlap	Float	
Volume	Float	
Output	Type	Description
Return Value	UFoxTransitionSoundAction*	
NewComponent	UAudioComponent*	Component reference

Use Case: Crossfade action — smooth transition between sound assets.

3.4 Play Sound With Fade In

Provides 3.4 Play Sound With Fade In functionality.

Input	Type	Description
WorldContext	Object Reference	
Sound	Sound Base	
FadeInDuration	Float	Time in seconds
Volume	Float	
bIs2D	Bool	Boolean flag
Output	Type	Description
Return Value	UAudioComponent*	

Use Case: Crossfade action — smooth transition between sound assets.

3.5 Stop Sound With Fade

Provides 3.5 Stop Sound With Fade functionality.

Input	Type	Description
AudioComponent	UAudioComponent*	Component reference
FadeOutDuration	Float	Time in seconds

(none)

Use Case: Crossfade action — smooth transition between sound assets.

4. Spatial (4 Nodes)

4.1 Play Sound At Actor

Provides 4.1 Play Sound At Actor functionality.

Input	Type	Description
WorldContext	Object Reference	
Sound	Sound Base	
Actor	Actor Reference	Target actor reference
Volume	Float	
Pitch	Float	
Output	Type	Description
Return Value	UAudioComponent*	

Use Case: Spatial sound — play or attach sound to actors with distance falloff.

4.2 Play Sound Following Actor

Provides 4.2 Play Sound Following Actor functionality.

Input	Type	Description
WorldContext	Object Reference	
Sound	Sound Base	
Actor	Actor Reference	Target actor reference
AttachRule	EFoxSoundAttachRule	
Output	Type	Description
Return Value	UAudioComponent*	

Use Case: Spatial sound — play or attach sound to actors with distance falloff.

4.3 Get Nearest Sound Source

Provides 4.3 Get Nearest Sound Source functionality.

Input	Type	Description
WorldContext	Object Reference	
Origin	Vector	
Radius	Float	Radius in units
Output	Type	Description
Return Value	Bool	
NearestComponent	UAudioComponent*	Component reference
Distance	Float	

Use Case: Spatial sound — play or attach sound to actors with distance falloff.

4.4 Set Sound Range

Provides 4.4 Set Sound Range functionality.

Input	Type	Description
AudioComponent	UAudioComponent*	Component reference
MinDistance	Float	
MaxDistance	Float	

(none)

Use Case: Spatial sound — play or attach sound to actors with distance falloff.

5. Utility (7 Nodes)

5.1 Play Sound Chain

Provides 5.1 Play Sound Chain functionality.

Input	Type	Description
WorldContext	Object Reference	
Sounds	Array of Sound Base	
DelayBetween	Float	
Volume	Float	
Output	Type	Description
Return Value	UFoxPlaySoundChainAction*	

Use Case: Utility helper — supporting function for the toolkit.

5.2 Play Random Sound At Location

Provides 5.2 Play Random Sound At Location functionality.

Input	Type	Description
WorldContext	Object Reference	
Sounds	Array of Sound Base	
Location	Vector	World-space location
Volume	Float	
PitchVariation	Float	
Output	Type	Description
Return Value	Sound Base	

Use Case: Utility helper — supporting function for the toolkit.

5.3 Play Sound With Random Pitch

Provides 5.3 Play Sound With Random Pitch functionality.

Input	Type	Description
WorldContext	Object Reference	
Sound	Sound Base	
Location	Vector	World-space location
MinPitch	Float	
MaxPitch	Float	

(none)

Use Case: Utility helper — supporting function for the toolkit.

5.4 Play Sound Looped With Delay

Provides 5.4 Play Sound Looped With Delay functionality.

Input	Type	Description
WorldContext	Object Reference	
Sound	Sound Base	
Delay	Float	
Volume	Float	
Output	Type	Description
Return Value	UAudioComponent*	

Use Case: Utility helper — supporting function for the toolkit.

5.5 Stop All Sounds On Actor With Fade

Provides 5.5 Stop All Sounds On Actor With Fade functionality.

Input	Type	Description
Actor	Actor Reference	Target actor reference
FadeOutDuration	Float	Time in seconds

(none)

Use Case: Utility helper — supporting function for the toolkit.

5.6 Get Playing Sound Count

Provides 5.6 Get Playing Sound Count functionality.

Input	Type	Description
Actor	Actor Reference	Target actor reference
Output	Type	Description
Return Value	Int	

Use Case: Utility helper — supporting function for the toolkit.

5.7 Is Any Sound Playing On Actor

Provides 5.7 Is Any Sound Playing On Actor functionality.

Input	Type	Description
Actor	Actor Reference	Target actor reference
Output	Type	Description
Return Value	Bool	

Use Case: Utility helper — supporting function for the toolkit.

6. Volume (4 Nodes)

6.1 Fade Volume For Tag

Provides 6.1 Fade Volume For Tag functionality.

Input	Type	Description
WorldContext	Object Reference	
Tag	Name	Gameplay tag name
TargetVolume	Float	
Duration	Float	Time in seconds
Output	Type	Description
Return Value	UFoxFadeVolumeForTagAction*	

Use Case: Volume mixer — adjust volume globally or by audio tag.

6.2 Set Volume For Tag

Provides 6.2 Set Volume For Tag functionality.

Input	Type	Description
WorldContext	Object Reference	
Tag	Name	Gameplay tag name
Volume	Float	

(none)

Use Case: Volume mixer — adjust volume globally or by audio tag.

6.3 Mute Unmute Tag

Provides 6.3 Mute Unmute Tag functionality.

Input	Type	Description
WorldContext	Object Reference	
Tag	Name	Gameplay tag name
bMute	Bool	Boolean flag

(none)

Use Case: Volume mixer — adjust volume globally or by audio tag.

6.4 Set Master Volume

Provides 6.4 Set Master Volume functionality.

Input	Type	Description
WorldContext	Object Reference	
Volume	Float	

(none)

Use Case: Volume mixer — adjust volume globally or by audio tag.

Implementation Notes

- Subsystem owns 4 UAudioComponents: MusicA, MusicB (for crossfade ping-pong), Ambient, CombatMusic.
- Crossfade ping-pongs between MusicA and MusicB to avoid glitches.
- Region stack: TArray stack, max depth 8.
- Combat/Danger state stored per-region, restored on region pop.
- All crossfades use FTimerDelegate with lerp, not AdjustVolume (more control).
- Register doesn't load sounds — uses TSoftObjectPtr, loads on Enter.