# Wwise Unreal Plug-in

# Requirements

## Unreal Engine

Each release of the Unreal Wwise plug-in is customized for a specific version of the Unreal Engine. Be sure to use matching versions of the Unreal Engine and Wwise plug-in.

This plug-in is intended for use with Unreal Engine 4.17.

# Wwise

This plug-in is based on Wwise 2016.2.5.6121; while it is possible to use an alternate version of the Wwise SDK, modifications to the plug-in code may be necessary.

# Platforms

This plug-in has been tested on Windows, Mac, Linux (SteamOS), Xbox One, PlayStation 4, Android, and iOS. Modifications may be required to target other platforms.

# Development Environment Setup

Please refer to the UE4 documentation:

- "Downloading Unreal Engine Source Code"
- "Setting Up Visual Studio for UE4"
- "Building Unreal Engine from Source"

# Wwise Unreal Plug-in

# Installation

Before proceeding, review the **Requirements** page and make sure the appropriate versions of the Unreal Engine, Unreal Wwise plug-in, and Wwise are used.

This UE4 integration package contains files related to the Wwise plug-in only; it does *not* contain the entire Unreal Engine source code, nor the Wwise SDK.

# Installation Procedure

There are two ways of installing the Unreal Wwise plug-in: either as a game plug-in or as an engine plug-in - not both.
**NOTE:** For more information on the difference between engine and game plug-ins (sometimes referred to as "installed" plug-ins), please refer to the Unreal Engine Wiki.

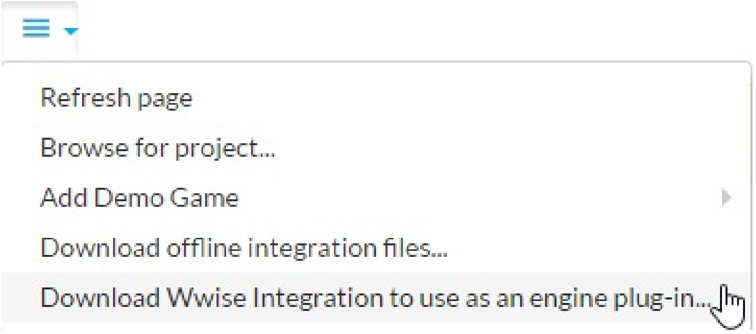
## *Installing the Wwise Plug-in as a Game Plug-in*

Installing as a game plug-in is handled by the Wwise Launcher. Please open the Launcher to the **UNREAL ENGINE** tab and find your Unreal project. If it's not there, then click **Browse for project...** to find the `UPROJECT` file and add the project to the Launcher. Once it's listed, click **Integrate Wwise into Project...** for the Unreal project. The Launcher will guide you to specify all the necessary information to successfully integrate the Wwise plug-in into your project. Alternatively, you can accomplish the same thing using offline integration files, which are downloaded via the **UNREAL ENGINE** tab menu.

## *Installing the Wwise Plug-in as an Engine Plug-in*

**NOTE:** Installing the Wwise plug-in for Unreal as an engine plug-in is intended for expert users only.

To install as an engine plug-in, you must first download the integration from the Launcher. Go to the **UNREAL ENGINE** tab and select the **Download Wwise Integration to use as an engine plug-in...** option from the tab's menu.


Refresh page
Browse for project...
Add Demo Game
Download offline integration files...
Download Wwise Integration to use as an engine plug-in...

Then, follow these steps:

1. Copy the "Wwise" UE4 Integration folder to the "Plugins" folder. Copy the "Wwise" UE4 Integration folder to `…/<UE4 installation directory>/Engine/Plugins`.
2. Since the Wwise UE4 integration may need to be rebuilt on the fly by the Unreal Editor during the packaging process, several folders from the Wwise SDK installation folder must be copied into the "Wwise" plug-in folder hierarchy. The following folders are required:

- Source: `C:\Program Files (x86)\Audiokinetic\Wwise 2016.2.5.6121\SDK\include\*.*`
- Destination: `.../Plugins/Wwise/ThirdParty/include`

- Source: `C:\Program Files (x86)\Audiokinetic\Wwise 2016.2.5.6121\SDK\Win32_*\*.*`
- Destination: `.../Plugins/Wwise/ThirdParty/Win32_*`

- Source: `C:\Program Files (x86)\Audiokinetic\Wwise 2016.2.5.6121\SDK\x64_*\*.*`
- Destination: `.../Plugins/Wwise/ThirdParty/x64_*`

- Source: `C:\Program Files (x86)\Audiokinetic\Wwise 2016.2.5.6121\SDK\Mac\*.*`
- Destination: `.../Plugins/Wwise/ThirdParty/Mac`

You then need to open your project and go to **Edit** > **Plugins** > **Audio** and select the **Enable** check box under "Wwise Unreal Engine 4 integration".

For Android, iOS, Linux, PlayStation 4, and Xbox One, the libraries must also be copied:

- Source: `C:\Program Files (x86)\Audiokinetic\Wwise 2016.2.5.6121\SDK\<Your Platform>\*.*`
- Destination: `.../Plugins/Wwise/ThirdParty/<Your Platform>`

# Wwise Unreal Plug-in

# Building the plug-in

## Preprocessor Defines

The following preprocessor defines are introduced to control the integration features:

- `AK_USE_UNREAL_IO` Located in `…\Plugins\Wwise\Source\AkAudio\AkAudio.Build.cs`.
  This setting, which is enabled by default, causes all IO requests from the Wwise sound engine to be routed in the Unreal IO system. When not enabled, the standard Wwise low-level I/O is used.

- `AK_SOUNDFRAME`
  Located in `…\Plugins\Wwise\Source\AkAudio\AkAudio.Build.cs`.
  This setting provides basic SoundFrame support to activate Unreal Editor features, such as Radius display, that require communication with the Wwise Authoring Application. Enabled by default, except in the Ship configuration. This option is not available on Mac OS X.

# Building the Unreal Wwise plug-in from source

If the installation steps outlined in **Installation** have been followed correctly, rebuilding the Unreal Wwise plug-in is done the same way as rebuilding the Unreal Engine. For a more detailed description of this process, please consult the Unreal Engine documentation.

For more information on plugins with source code, please refer to the Unreal Engine documentation.

# Wwise Unreal Plug-in

# Using the Integration

- **Initial Setup** How to set up using Wwise after building Unreal with the integration.
- **Workflow**
  A basic workflow to get started.
- **Sample Game**
  A quick overview of the UnrealWwiseDemo game demonstrating the integration features.
- **Features**
  Survey of the features introduced by the integration.
- **Frequently Asked Questions**
  Answers to the most frequently asked questions.

Generated by doxygen 1.6.3

# Wwise Unreal Plug-in » Using the Integration

# Initial Setup

## Initialization settings

**IMPORTANT:** All settings related to the Wwise Integration can be found in the Unreal Project Settings, under the Wwise section of the plug-ins section.

- **Max Simultaneous Reverb Volumes** The maximum number of Ak Reverb Volumes that can affect a sound. Setting this to zero disables all Ak Reverb Volumes in game.

- **Wwise Windows Installation Path**
  The location of the Wwise Authoring Application on your Windows development machine. This option will need to be updated when a new version of the Wwise Authoring Application is required by the integration changes.

- **Wwise Mac Installation Path**
  The location of the Wwise Authoring Application on your Mac OS X development machine. This option will need to be updated when a new version of the Wwise Authoring Application is required by the integration changes.

- **Wwise Project Path**
  The location of the Wwise project for the UE4 game. The Wwise integration requires this path to create the Wwise assets required for the game using the **Generate SoundBanks** function within the Unreal Content Browser (or Build menu). This path is relative to the editor's executable directory (usually `<UE4_installation_directory>/Engine/Binaries/Win64`), as given by `FPlatformProcess::BaseDir()` in the Unreal Engine.

If these paths are not correctly set, Unreal will fail to generate the Wwise SoundBanks required for the game.

# Initializing the SoundEngine

The SoundEngine initialization steps are performed in the `FAkAudioDevice::EnsureInitialized()` method. In this method, the memory, streaming, IO, sound engine, platform, music engine, and communication settings can be configured. This is also where plug-ins can be registered. To use a licensed plug-in, uncomment the corresponding line of code right after the call to `AK::SoundEngine::RegisterAllBuiltInPlugins()`.

For more information on SoundEngine initialization, refer to the Initialize the Different Modules of the Sound Engine section in the Wwise SDK documentation.

# Wwise Unreal Plug-in » Using the Integration

# Workflow

Here is a sample workflow using the integration:

# 1) Create a Wwise project

Create a Wwise project for this game using the Wwise Authoring Application. Some Events are created, but no SoundBanks need to be created yet.

## 2) Add AkAudioEvent objects to Unreal packages

In the Unreal Content Browser, add **AkAudioEvent** objects to packages.

## 3) Reference AkAudioEvent objects in a level

**AkAudioEvent** objects can now be referenced in a level, such as in using Wwise-specific **Blueprint Functions** or **AkAmbientSound** objects.

## 4) Add AkAudioBank objects

The Wwise assets associated with **AkAudioEvent** object can be distributed across any number of **AkAudioBank** objects. Using the Unreal Content Browser, add **AkAudioBank** objects as required.

## 5) Set the RequiredBank property for each AkAudioEvent

The `RequiredBank` property of each **AkAudioEvent** determines which **AkAudioBank** its content will be stored in. This property must be set for each **AkAudioEvent** used in the game.

## 6) Generate SoundBanks (for play-in-editor)

From the **Build** menu, select the **Generate SoundBanks** option, and then select all banks to be generated, and click **OK**. The required SoundBanks are generated. After the banks are generated at least once, it is possible to play sounds in the editor.

**Note:** The first time banks are generated, you need to click **Refresh All Banks** by right-clicking on one of the **AkAudioBank** objects.

Banks generated from the Unreal Editor can then be added to source control. This allow individuals without the Wwise application and project on their machine to still be able to work in the Unreal editor using existing BNK files from the game's Content folder. They could hook the **AkAudioEvent** objects at different places in the level, and they would be able to run the game.

When new banks are generated, click **Refresh All Banks** in the Content Browser's **AkAudioBank** to load the newly created banks.

# Wwise Unreal Plug-in » Using the Integration

# Sample Game

A sample game containing the Wwise UE4 Integration is available to download in the Wwise Launcher. Please refer to the Launcher.

The game's map provides several demonstration stations.

# Generation of the SoundBanks

**IMPORTANT:** Generate SoundBanks for this project from the WwiseDemoGame Unreal Editor:

1. Open the WwiseDemoGame in the Unreal Editor.
2. From the toolbar, click the arrow next to **Build** and select **Generate SoundBanks...** from the **Audiokinetic** category.
3. Select all the SoundBanks and the desired platforms and click **OK**.

On completion, the resulting content is visible in *.../WwiseDemoGame/Content/WwiseAudio*.

# Map Contents

## *AkEvent Animation Notify*

An Animation Notify can be used to post AkEvents to the SoundEngine. A demonstration of its use can be seen in the FPP_Fire animation.

## *Ambient Demo*

This area of the map demonstrates the use of the **AkAmbientSound** actor. The `AmbientNoise_Spatialized` ambient emitter in the level is started by a **Start All Ambient Sounds** node in the Level Blueprint. To help locate the actor, its attenuation sphere is drawn in yellow when playing the game.

The Event used by this emitter is part of the `AmbientBank` SoundBank, which is automatically loaded thanks to the **Auto Load** flag set on its AkAudioBank asset.

This area also helps to demonstrate the occlusion feature. By navigating the player pawn behind the wall situated close to the ambient sound (making sure to stay within the yellow zone), or behind the big box in the sphere, the effects of occlusion on the sound can be heard. Occlusion is enabled on the sound by setting it in the in the "Ambient Sound handling" (red) section of the Blueprint level.

## *Matinee Demo*

This section of the map demonstrates the use of Event and RTPC tracks in the MatineeDemoData asset. Opening the matinee sequence in the Matinee editor shows that the Event track plays a drum beat targeting the moving cube, while the `DrumKitModulation` game parameter is driven by the RTPC track.

## *RTPC Demo*

This section of the map demonstrates use of the **Set RTPCValue** node targeting an actor, in the Level Blueprint (in the green comment sections). The mouse scroll wheel (the up and down D-pad buttons on a gamepad, and a two-finger swipe on a touch screen) is tied to a variable that is fed as the "Velocity" Game Parameter, which controls the pitch of the `VelocityLoop` sound in the Wwise project.

The "Create RTPC button Event dispatchers" section of the level Blueprint also shows how to manually load and unload the VelocityBank (whose Auto-Load is unselected), using the **Load Bank** and **Unload Bank** Blueprint nodes. It also demonstrates how to post an Event that targets an actor, using the **Post Event** node.

## Reverb Demo

This section of the map demonstrates use of `AkReverbVolume`. Inside the spherical cave, an `AkReverbVolume` is used to add a Reverb Effect to the weapon sound. Note that the actor emitting the sound can be set to ignore the reverb volumes feature. This is done in the "Variable initializations" section of the Blueprint level.

## Switch Demo

This section of the map demonstrates how to set a Switch using blueprints (in the "Switch Logic" section of the Blueprint level). Using the button (see instructions on the wall in front of the button) posts the Event to the SoundEngine. Pressing L or H on the keyboard (D-Pad left and right on a gamepad, three and four finger taps on a touch screen) changes the switch value.

# Wwise Unreal Plug-in » Using the Integration

# Features

The following sections give an overview of the elements that the integration brings to Unreal.

- **Unreal Objects** Unreal Objects introduced to expose base Wwise concepts.
- **Editor Integration**
  SoundFrame-enabled workflow enhancements to the Editor.
- **Blueprint Functions**
  Blueprint functions available in visual scripting.
- **Animation**
  Animation Notify for Events.
- **Level Sequencer**
  Sequencer Tracks for Events and RTPC.
- **Matinee**
  Matinee Tracks for Events and RTPC.
- **Occlusion**
  Occlusion support.

# Unreal Objects

## AkAudioEvent

Unreal object representing a Wwise Event. Can be created either by **drag-and-drop** from the Wwise Picker or by right-clicking in the Unreal Content Browser. Its name should exactly match the name of an Event in the Wwise Project.

- Properties
  - **Required Bank**: Specifies the SoundBank that contains this Event along with its structures and media.
- Unreal Content Browser Context Menu Options
  - **Play Event**: Posts the Event.
  - **Stop Event**: Stops all currently playing Events.

# AkAuxBus

Unreal object representing a Wwise Auxiliary Bus. Can be created either by **drag-and-drop** from the Wwise Picker, or by right-clicking in the Unreal Content Browser. Its name should exactly match the name of an Auxiliary Bus in the Wwise Project. This allows to add plug-in media to a SoundBank from within the Unreal Editor.

- Properties
    - **Required Bank**: Specifies the SoundBank that contains this Auxiliary Bus along with its associated plug-in media.

# AkAudioBank

Unreal object representing a Wwise SoundBank. Can be created by right-clicking in the Unreal Content Browser.

- Properties
  - **Auto Load**: When enabled, automatically loads and unloads the SoundBank along with the package referencing it. **Note:** Enabling this flag does not instantly load the bank in the editor; it is only applied on the next package load.
- Unreal Content Browser Context Menu Options
  - **Generate Selected SoundBank(s)...**: Launches a dialog window where a SoundBank Generation operation can be performed on a list of SoundBanks.
  - **Load Bank**: Loads this SoundBank.
  - **Unload Bank**: Unloads this SoundBank.
  - **Clear Banks**: Unloads all currently loaded SoundBanks, including the Init bank.
  - **Load Init Bank**: Loads the Init Bank. This Bank must be loaded prior to any other SoundBank.
  - **Refresh All Banks**: Stops all sounds; unloads and then loads all currently loaded SoundBanks.

# AkAmbientSound

`AkAmbientSound` is an `AActor` class used in the same way as the `AAmbientSound` object that is supplied with the default Unreal Audio system. Its playback is controlled either via its own object Blueprint functions, or by using the global helper functions Start All Ambient Sounds and Stop All Ambient Sounds.

- Properties
    - **Ak Audio Event**: The **AkAudioEvent** that is posted when the **AkAmbientSound** object is instructed to start playing.
    - **Stop When Owner Is Destroyed**: When enabled, the Event is stopped automatically when the AkAmbientSound is destroyed.
    - **Attenuation Scaling Factor**: If the Ambient Sound uses 3D attenuation in Wwise, this property allows modifying the attenuation computations on this ambient sound to simulate sounds with a larger or smaller area of effect.
- Blueprint Functions
    - **Start All Ambient Sound**: Starts the playback of all ambient sounds.
    - **Start Ambient Sound**: Starts the playback of the selected ambient sound.
    - **Stop All Ambient Sound**: Stops the playback of all ambient sounds.
    - **Stop Ambient Sound**: Stops the playback of the selected ambient sound.

# AkReverbVolume

`AkReverbVolume` is an `AVolume` class is used in a similar way to the `AReverbVolume` object that is supplied with the default Unreal Audio system. It can be spawned from any `Brush` in the Editor. The reverb effect is obtained by assigning a Wwise Auxiliary Bus to the volume, and routing all `AkComponents` entering this volume to the associated Wwise Auxiliary Bus. If there is volume overlap, a `Priority` property is used to determine which Auxiliary Buses the target `AkComponent` is routed to. A temporal fade in/out Effect is applied to the level of the Auxiliary Bus when entering/exiting a `AkReverbVolume`.

- Properties
    - **Aux Bus Name**: Name of the Wwise Auxiliary Bus assigned to this volume
    - **Send Level**: Maximum Send Level associated with the Wwise Auxiliary Bus
    - **Fade Rate**: Rate at which to fade in/out the SendLevel of the current Reverb Volume when entering/exiting it, in percentage per second (0.2 will make the fade time 5 seconds).
    - **Priority**: The precedence in which the AkReverbVolumes will be applied. In the case of overlapping volumes, only the ones with the highest priority are chosen (the number of simultaneous AkReverbVolumes is configurable in the Unreal Editor Project Settings under Plugins > Wwise). If two or more overlapping AkReverbVolumes have the same priority, the chosen AkReverbVolume is unpredictable.

# AkComponent

`AkComponent` is derived from `USceneComponent` and represents an active Wwise event.

- Properties
    - **Stop when Attached to Destroyed**: When set to `true`, playback of the component stops if the owning actor is destroyed.
- Blueprint Functions
    - **Post Ak Event**: Starts playback of the specified Event.
    - **Post Trigger**: Posts a Trigger to the Event associated with the component.
    - **Set Occlusion Refresh Interval**: Sets the time interval at which the UAkComponent performs occlusion calculations. Set to 0 to turn off occlusion on the component.
    - **Set RTPC Value**: Sets the value of the RTPC of the Event associated with the component.
    - **Set Stop when Owner Destroyed**: Sets the `StopWhenOwnerDestroyed` value on the component.
    - **Set Switch**: Sets a Switch of the Event associated with the component.
    - **Stop**: Stops playback of the Event associated with the component.
    - **Use Reverb Volumes**: Sets whether the component is influenced by `AAkReverbVolumes` or not.

# Editor Integration

## Event Drag-and-Drop

Events and Auxiliary Busses can be dragged directly from the Wwise Picker into the Unreal Content Browser to create corresponding **AkAudioEvent** and **AkAuxBus** objects:

1. In the Unreal Editor, navigate to the Event or Auxiliary Bus using the Wwise Picker.
2. In the Unreal Editor, select the target Package in the Unreal Content Browser.
3. Drag the Event or Auxiliary Bus into the Unreal Content Browser: a corresponding **AkAudioEvent** or **AkAuxBus** object is created.

## Radius Visualization

Some workflow enhancements in the Unreal Editor are enabled through SoundFrame when running the Wwise Authoring Application alongside the Unreal Editor on the same machine.

With SoundFrame enabled and the Wwise Authoring Application running, the radius of the relevant assets are drawn when an **AkAmbientSound** is selected. If changes are made in Wwise Authoring, the radius is updated in real time:

1. Select an **AkAmbientSound** in the Actor Browser.
2. In the Wwise Authoring Application, make changes to any object associated with the Event assigned to the **AkAmbientSound**.
3. The corresponding sphere in the Unreal Editor resizes at the same time.

If the Wwise Authoring Application is not running on your machine, it is still possible to see the attenuation sphere in Unreal. In your Wwise project settings, under the SoundBanks tab, make sure to select **Generate Per Bank Metadata File**, **Generate JSON Metadata**, and **Max Attenuation**. The next time the SoundBanks are generated in the Unreal Editor, the Max Attenuation information will be added to all `UAkAudioEvents`, and the attenuation spheres will be visible in your level viewport.

---

# Blueprint Functions

Several Wwise-specific global functions are exposed to scripting; the following are available in the Audiokinetic category:

# Get Ak Component

Obtains an **AkComponent** that is attached to and follows a specified component. Several methods are available directly on the **AkComponent** : see **AkComponent Blueprint Functions**.

# Post Event At Location

Posts a Wwise Event at a specified location. This is a fire-and-forget sound, created on a temporary Wwise Game Object with no corresponding **AkComponent**.

# Set RTPC Value

Sets the value of a Game Parameter, optionally targeting the root component of a specified actor.

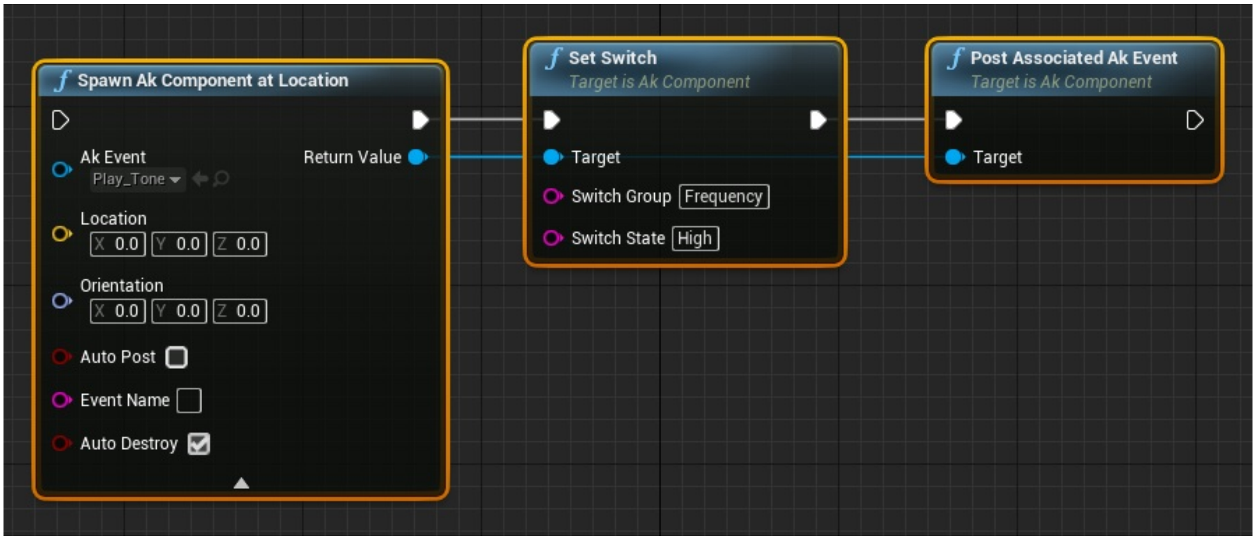
# Set State

Sets the active State for a given State Group.

# Spawn Ak Component at Location

Creates a new AkComponent at the specified location. By default, the component will automatically be destroyed once the Event it is playing is done. Parameter `Auto Post` controls whether to post the event immediately upon the component's creation (Default value is `false`). Advanced parameter `Auto Destroy` controls whether to destroy the component when the first Event posted on this component is done (Default value is `true`).

This Blueprint node is especially useful if you wish to set a Switch on a fire-and-forget sound. This behavior can be accomplished by disabling `Auto Post`, setting a Switch on the spawned Ak Component, and then posting the Event, like so:



Ak Event Play_Tone
Location X 0.0 Y 0.0 Z 0.0
Orientation X 0.0 Y 0.0 Z 0.0
Auto Post □
Event Name
Auto Destroy ☑

Switch Group Frequency
Switch State High

# Stop All

Stops all currently playing sounds.

# Other

Other functions are available in sub-categories:

- **Actor Blueprint Functions**
- **AkAmbientSound Blueprint Functions**
- **AkComponent Blueprint Functions**
- **SoundBank Blueprint Functions**
- **Debug Blueprint Functions**

Generated by doxygen 1.6.3

# Wwise Unreal Plug-in » Using the Integration » Features » Blueprint Functions

# Actor Blueprint Functions

These Blueprint functions target the root component of an Actor. If no **AkComponent** is attached to the Actor's root component, one is created.

# Post Event

Posts a Wwise Event attached to and following the root component of the specified Actor.

## Post Trigger

Posts a Trigger to Wwise, targeting the root component of a specified Actor.

# Set Occlusion Refresh Interval

Sets the time interval at which the **AkComponent** attached to the root component performs occlusion calculations. Set to 0 to turn off occlusion on the component.

# Set Switch

Sets the active Switch for a given Switch Group, targeting the root component of a specified Actor.

# Set Output Bus Volume

Set the output bus volume (direct) to be used for the specified game object. The Bus Volume value is a number ranging from 0.0f to 1.0f.

## Stop Actor

Stops all sounds on the given Actor.

# Use Reverb Volumes

Sets whether the **AkComponent** attached to the root component is influenced by **AkReverbVolume** or not.

# Wwise Unreal Plug-in » Using the Integration » Features » Blueprint Functions

# AkAmbientSound Blueprint Functions

Several Wwise-specific actions can be executed on **AkAmbientSound** actors; these are available in the Ambient Sound category.

# Start All Ambient Sounds

Starts playback of all **AkAmbientSound** present in the map.

# Start Ambient Sound

Starts playback of the **AkAmbientSound**.

# Stop All Ambient Sounds

Stops playback of all **AkAmbientSound** present in the map.

# Stop Ambient Sound

Stops playback of the **AkAmbientSound**.

# Wwise Unreal Plug-in » Using the Integration » Features » Blueprint Functions

# AkComponent Blueprint Functions

Several Wwise-specific actions can be executed on **AkComponent** scene components; these are available in the Ak Component category.

# Get Attenuation Radius

Returns this **AkComponent**'s effective attenuation radius (ScalingFactor * MaxAttenuation).

# Post Ak Event

Posts the specified **AkAudioEvent** in Wwise.

# Post Associated Ak Event

Posts this **AkComponent**'s internal **AkAudioEvent** in Wwise.

# Post Trigger

Posts a trigger on the associated **AkComponent**.

# Set Active Listeners

Sets the listener mask on the associated **AkComponent**. For more information on the listener mask, please refer to the Wwise SDK documentation.

# Set Occlusion Refresh Interval

Sets the time interval at which the **AkComponent** performs occlusion calculations. Set to 0 to turn off occlusion on the component.

# Set Output Bus Volume

Set the output bus volume (direct) to be used for the specified game object. The Bus Volume value is a number ranging from 0.0f to 1.0f.

# Set RTPC Value

Sets the value of a Game Parameter for the associated **AkComponent**.

# Set Stop when Owner Destroyed

Sets the StopWhenOwnerDestroyed value on the corresponding **AkComponent**.

# Set Switch

Sets a Switch Group to a given Switch on the associated **AkComponent**.

# Stop

Stops the **AkAudioEvent** associated with the **AkComponent** from playing.

# Use Reverb Volumes

Sets whether the **AkComponent** is influenced by **AkReverbVolume** or not.

# Wwise Unreal Plug-in » Using the Integration » Features » Blueprint Functions

# SoundBank Blueprint Functions

Several Wwise-specific actions can be executed on **AkAudioBank** scene components; these are available in the SoundBanks category.

# Clear Banks

Results in all currently loaded **AkAudioBank** being unloaded synchronously, **including the init SoundBank**.

# Load Bank

Loads a specified SoundBank.

# Load Banks

Loads a specified list of SoundBanks. If `SynchronizeSoundBanks` is set to `true`, only the SoundBanks in `Banks` will remain loaded in the SoundEngine. This means that any SoundBank that was previously loaded, and that is not found in the `Banks` array, will be unloaded.

# Load Init Bank

Loads the Init SoundBank.

# Unload Bank

Unloads a specified SoundBank.

# Wwise Unreal Plug-in » Using the Integration » Features » Blueprint Functions

# Debug Blueprint Functions

These Blueprint functions are used to help debug your game.

## Start Output Capture

Starts a Wwise output capture. The output file will be located in the same folder as the SoundBanks.

# Add Output Capture Marker

Adds a text marker in the output capture file.

## Stop Output Capture

Stops a Wwise output capture. The output file will be located in the same folder as the SoundBanks.

## Start Profiler Capture

Starts a Wwise Profiler capture. The output file will be located in the same folder as the SoundBanks.

# Stop Profiler Capture

Stops a Wwise Profiler capture. The output file will be located in the same folder as the SoundBanks.

# Animation

An Animation Notify is used to trigger Wwise Events in animation sequences.

The Blueprint for the Animation Notify is located here: `…/Plugins/Wwise/Content/AnimNotify_AkEvent`.uasset.

- To add an Animation Notify:
  1. In the Unreal Content Browser, open an animation sequence asset.
  2. Right-click the **Notifies** track in the animation sequence editor and select **Add notify > AkEvent** or **AkEventByName**.

- The animation notification has the following properties:
  - **Event** or **Event Name**: The **AkAudioEvent** to be posted.
  - **Attach Name**: Name of the socket that emits the **AkAudioEvent**; if not defined, the AkAudioEvent is emitted from the animation owner.
  - **Follow**: Whether the Event should either follow the mesh or be posted at a specific location.

---

Generated by doxygen 1.6.3

# Level Sequencer

The Sequence Editor within Unreal Engine 4 is a cinematic editing tool similar to Matinee. It allows users to add Tracks that can modify certain properties of Actors in a level.

For instructions on how to:

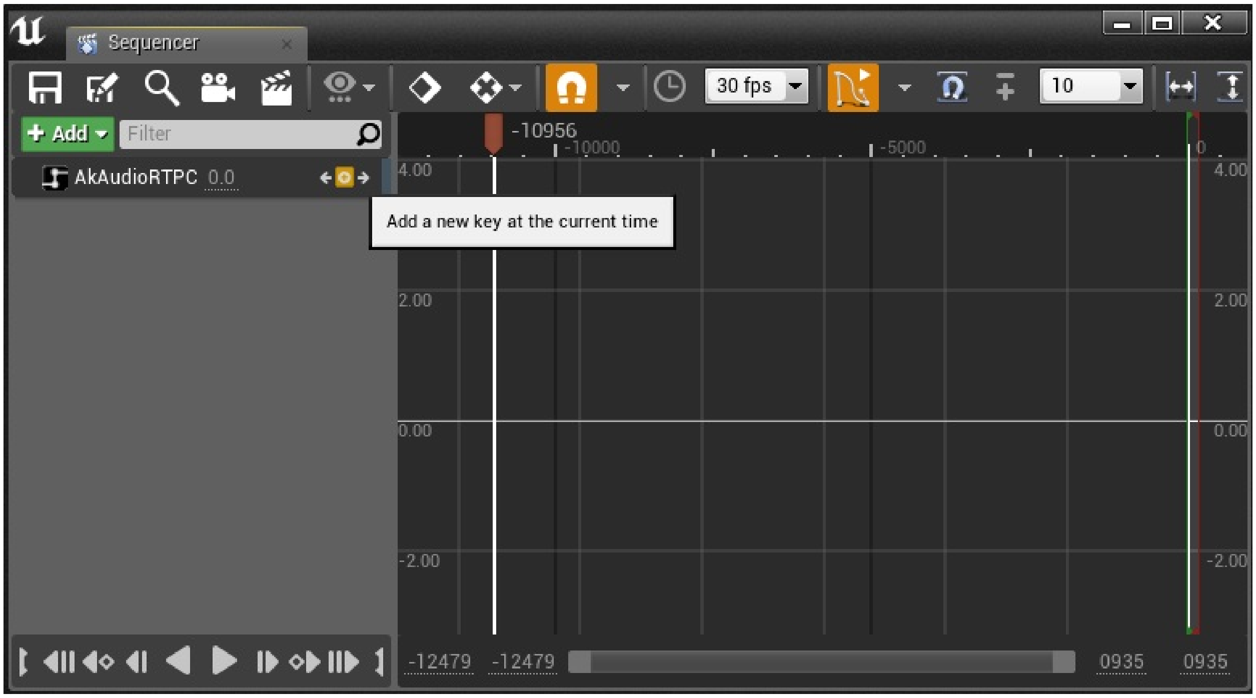
- create a Level Sequence, refer to https://docs.unrealengine.com/latest/INT/Engine/Sequencer/Overviev
- add Tracks to a Level Sequence, refer to https://docs.unrealengine.com/latest/INT/Engine/Sequencer/Overviev

# Wwise Level Sequencer Tracks

The Wwise Unreal Engine 4 Integration adds two Tracks: **AkAudioRTPC** and **AkAudioEvent**. The AkAudioRTPC Track is used to set Game Parameter values, and the AkAudioEvent Track is used to post Wwise Events.

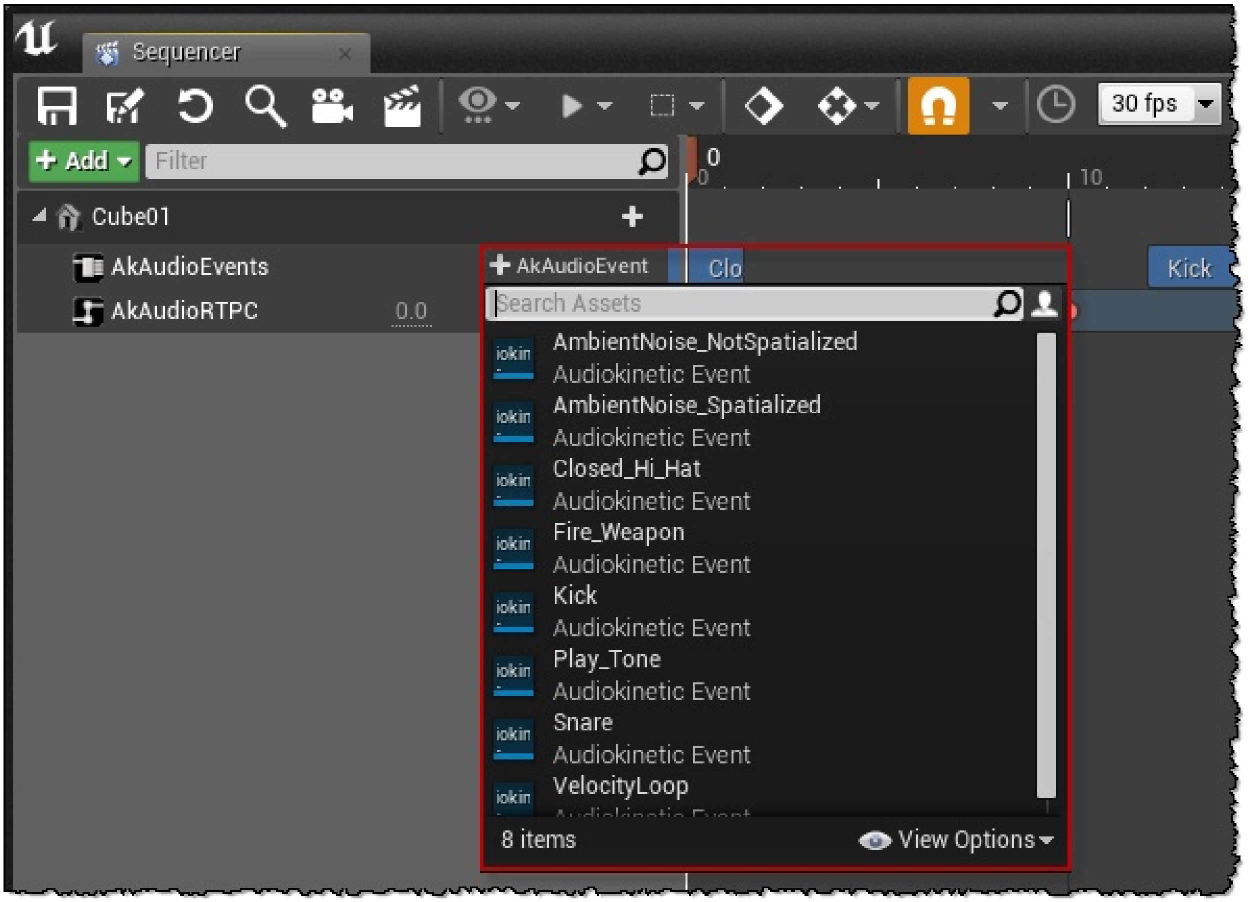
Both of these Tracks can be added as Master Tracks or associated to an Actor. When associated with an Actor, they perform their Wwise related functions on the `UAkComponent` attached to this Actor. When created as a Master Track, the AkAudioRTPC Track sets global RTPC values, and the AkAudioEvent Track posts Events on a "dummy" game object.

For **AkAudioRTPC** Tracks, Game Parameter curves can be modified using the built-in curve editor. Key frames can be added by placing the cursor at the appropriate location and pressing the Add New Key button (found on the right side of the listed Track).



**An AkAudioRTPC track in the Curve Editor view**

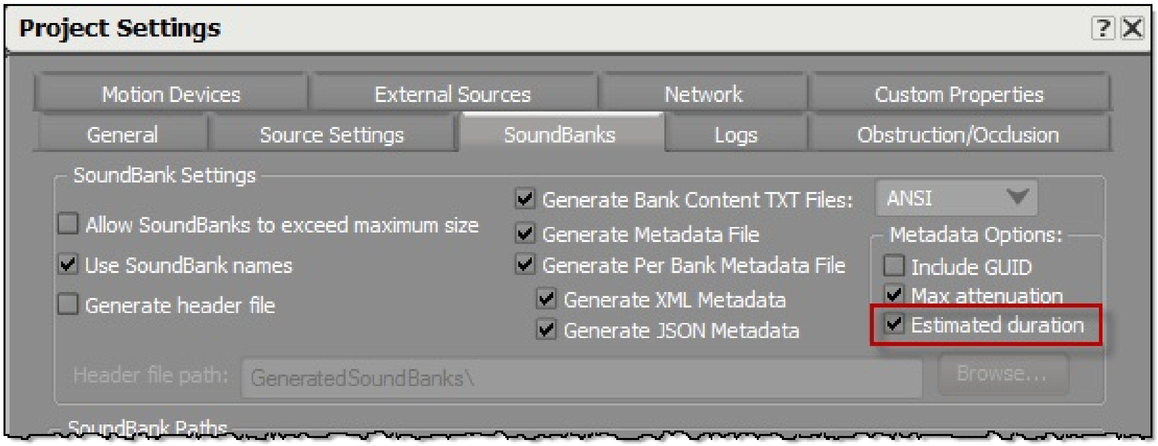
For **AkAudioEvent** Tracks, an `AkAudioEvent` can be added by placing the cursor at the appropriate location and pressing the **AkAudioEvent** button (found on the right side of the listed Track).



**Adding an AkAudioEvent to an AkAudioEvent Track**

# Wwise Project Setup

To ensure that Event lengths are properly represented in the AkAudioEvent Tracks, the Wwise project needs to be set up to estimate the duration of its audio Events and JSON metadata must be generated. To enable these project-wide settings, go to the Project Settings' SoundBanks tab. Enable the **Estimated duration** check box within the **Metadata Options** group box and enable the **Generate JSON Metadata** check box within the **SoundBank Settings** group box, as seen below.



SoundBank Settings

☐ Allow SoundBanks to exceed maximum size

☑ Use SoundBank names

☐ Generate header file

☑ Generate Bank Content TXT Files: ANSI

☑ Generate Metadata File

☑ Generate Per Bank Metadata File

☑ Generate XML Metadata

☑ Generate JSON Metadata

Metadata Options:

☐ Include GUID

☑ Max attenuation

☑ Estimated duration

Header file path: GeneratedSoundBanks\

**Required Project Settings for AkAudioEvent Tracks**

# Matinee

Two Wwise-specific Track types are introduced to Matinee:

# AkAudioEvent Track

The AkEvent Track triggers Wwise Events at particular points in the sequence. When placing a new key, the **AkAudioEvent** currently selected in the Unreal Content Browser is used.

# AkRTPC Track

The AkRTPC Track lets you control a Wwise Game Parameter using a curve in the sequence.

- Properties
    - **Param**: The name of the Game Parameter to control. This name should exactly match the name of a Game Parameter in the Wwise Project.

# Wwise Unreal Plug-in » Using the Integration » Features

# Occlusion

Basic occlusion is exposed in `UAkComponent::SetOcclusion()`. To enable occlusion on an Actor, use the **Set Occlusion Refresh Interval** function in the Blueprint Editor. A version for a UAkComponent is also available. If the refresh interval is set to 0, the occlusion check will never be performed.

Determining whether a listener is occluded from a source is done with a simple line-of-sight check. If the line of sight is blocked, the occlusion level calculation is started. This calculation maps the hit point on the obstacle to its bounding box, creating twelve points around the obstacle. To see if these "secondary paths" are also blocked, other line-of-sight tests are done. The occlusion sent to the SoundEngine is modulated from the number of "secondary paths" that are blocked.

A temporal fade method is also added to allow smooth transition between occlusion levels. To change the fade speed, change the `OCCLUSION_FADE_RATE` constant.

# Frequently Asked Questions

**Q:** Where are banks generated when I generate them from the Unreal editor or when cooking?
**A:** By default, they are generated in the following location:

```
UE4/[GameName]/Content/WwiseAudio/[Platform]
```

Some banks containing language assets may be in language-specific folders.

**Q:** I am using the new custom platforms feature in Wwise 2015.1, and my SoundBanks are not loading. Why?
**A:** With the custom platforms feature, the SoundBanks are generated in a sub-folder with the same name as your custom platform. You will need to modify the integration code to point to that folder. Open the file `…/Wwise/Source/AkAudio/Private/AkAudioDevice.cpp`, and search for the `FAkAudioDevice::SetBankDirectory` method. In this method, you will see the various default sub-folders for the supported platforms. Modify the one corresponding to your custom platform, and recompile the integration.

**Q:** When I generate banks from the Wwise application, I noticed that fewer banks are generated than when generating banks from the Unreal Editor. Why?
**A:** When importing a SoundBank definition file, Unreal may ask Wwise to generate banks that do not exist in the Wwise project. If you want the Unreal-generated bank to persist in the Wwise project, you can add the command-line argument `-Save` when importing the SoundBank definition file.

**Q:** When I try to use a plug-in in Unreal Engine 4, I get the following error: `LogAkAudio:Error: Plug-in not registered: 4163`. How do I use plug-ins?
**A:** In order for your plug-in to register itself with the SoundEngine, you need to include the plug-in's factory header (.h) file. To do so, add the necessary `include` in `…/Plugins/Wwise/Source/AkAudio/Private/AkAudioDevice.cpp`, under the

line saying

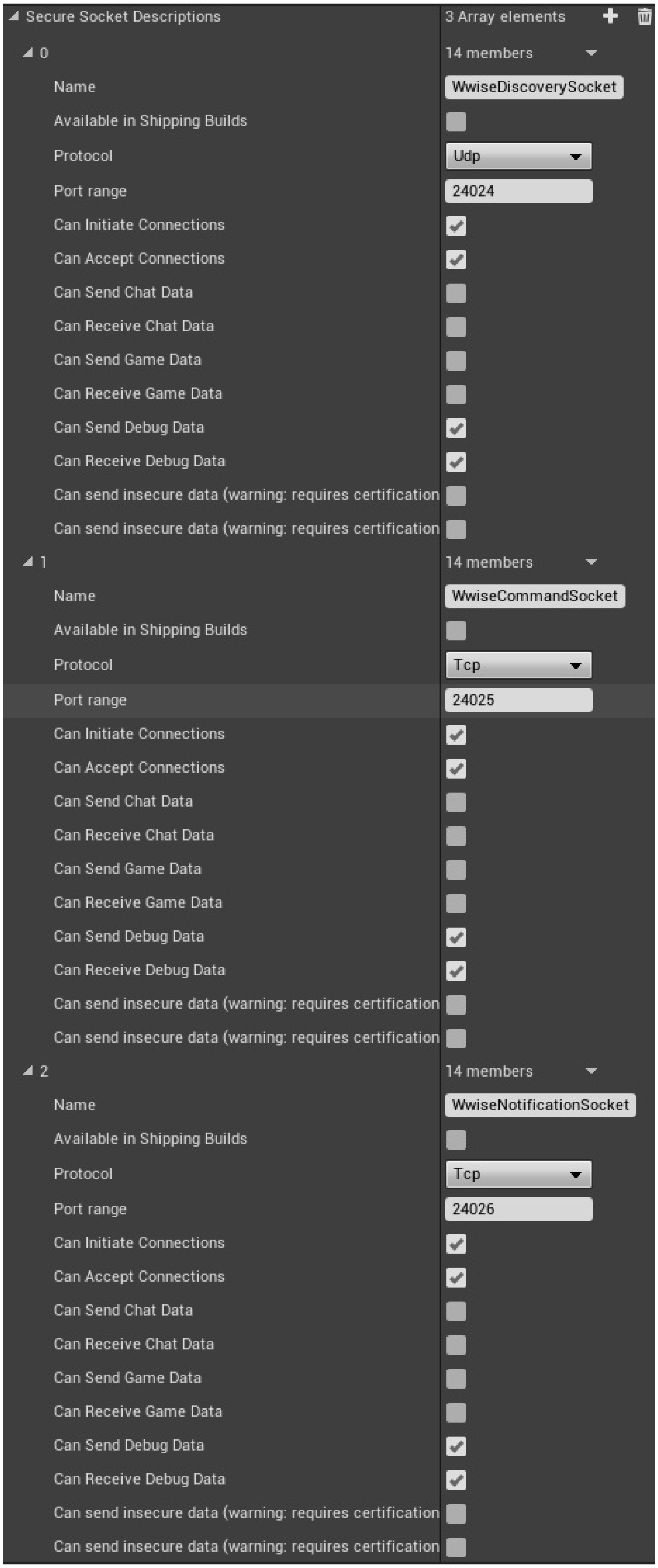
```
// Add additional plug-ins here.
```

You then need to link against the plug-in's library. To do so, add a call to the `AddWwiseLib` function for your plug-in in `…/Plugins/Wwise/Source/AkAudio/AkAudio.Build.cs`, near the other plug-ins. You then need to rebuild your game project from the Visual Studio solution or Xcode workspace.

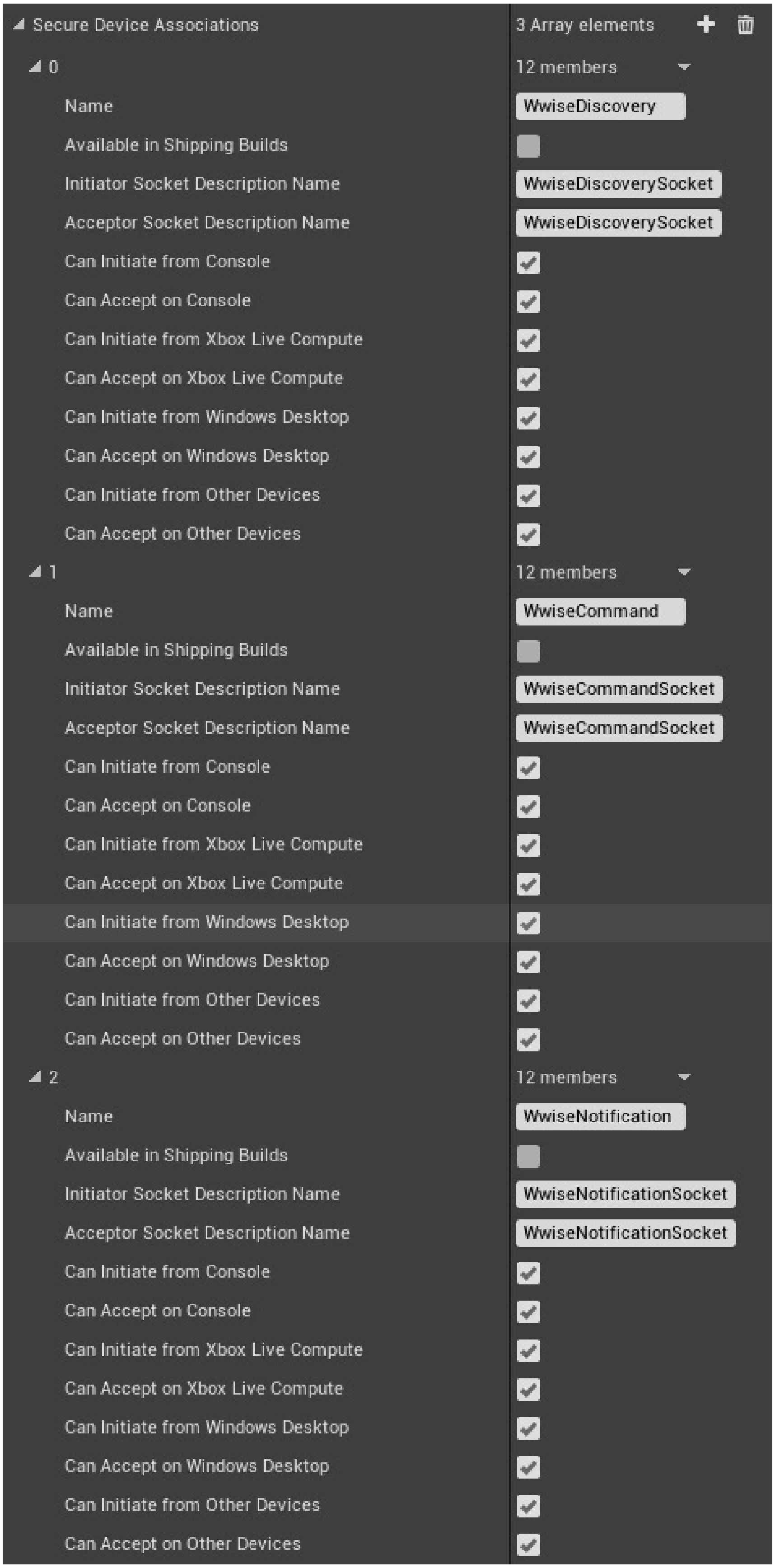
**Note:** A limitation in Unreal Engine 4 prevents from rebuilding an Unreal plug-in from a Content-only (Blueprint) project. Therefore, third-party Wwise DSP plug-ins are not available in a Content-only (Blueprint) project, as this method requires rebuilding the Wwise Unreal plug-in.

**Q:** Can I debug my code using the Wwise SoundEngine debug libraries?
**A:** By default, the Unreal Engine does not use the debug C Runtime Libraries, but Wwise uses them. When building in debug, this would cause linking errors. This is why, by default, the Wwise Unreal integration uses the profile libraries. If you wish to link against the debug Wwise libraries, simply set the `bDebugBuildsActuallyUseDebugCRT` variable to `true` in the following file:
`<UE4_ROOT>/Engine/Source/Programs/UnrealBuildTool/Configuration/Buil`

**Q:** I can't connect to my game running on an Xbox One, and I get the following error in the log: "Could not find Wwise network ports in AppxManifest. Network communication will not be available." How can I connect the Wwise Authoring Application to my game?
**A:** You should make sure to add the required Wwise network ports to your game's AppxManifest.xml. To do so, ensure the following Xbox One platform settings are filled in your project settings (Edit > Project Settings):



| | |
|---|---|
| ◢ Secure Socket Descriptions | 3 Array elements |
| ◢ 0 | 14 members |
| Name | WwiseDiscoverySocket |
| Available in Shipping Builds | ☐ |
| Protocol | Udp |
| Port range | 24024 |
| Can Initiate Connections | ☑ |
| Can Accept Connections | ☑ |
| Can Send Chat Data | ☐ |
| Can Receive Chat Data | ☐ |
| Can Send Game Data | ☐ |
| Can Receive Game Data | ☐ |
| Can Send Debug Data | ☑ |
| Can Receive Debug Data | ☑ |
| Can send insecure data (warning: requires certification | ☐ |
| Can send insecure data (warning: requires certification | ☐ |
| ◢ 1 | 14 members |
| Name | WwiseCommandSocket |
| Available in Shipping Builds | ☐ |
| Protocol | Tcp |
| Port range | 24025 |
| Can Initiate Connections | ☑ |
| Can Accept Connections | ☑ |
| Can Send Chat Data | ☐ |
| Can Receive Chat Data | ☐ |
| Can Send Game Data | ☐ |
| Can Receive Game Data | ☐ |
| Can Send Debug Data | ☑ |
| Can Receive Debug Data | ☑ |
| Can send insecure data (warning: requires certification | ☐ |
| Can send insecure data (warning: requires certification | ☐ |
| ◢ 2 | 14 members |
| Name | WwiseNotificationSocket |
| Available in Shipping Builds | ☐ |
| Protocol | Tcp |
| Port range | 24026 |
| Can Initiate Connections | ☑ |
| Can Accept Connections | ☑ |
| Can Send Chat Data | ☐ |
| Can Receive Chat Data | ☐ |
| Can Send Game Data | ☐ |
| Can Receive Game Data | ☐ |
| Can Send Debug Data | ☑ |
| Can Receive Debug Data | ☑ |
| Can send insecure data (warning: requires certification | ☐ |
| Can send insecure data (warning: requires certification | ☐ |



| | |
|---|---|
| Secure Device Associations | 3 Array elements |
| 0 | 12 members |
| Name | WwiseDiscovery |
| Available in Shipping Builds | □ |
| Initiator Socket Description Name | WwiseDiscoverySocket |
| Acceptor Socket Description Name | WwiseDiscoverySocket |
| Can Initiate from Console | ☑ |
| Can Accept on Console | ☑ |
| Can Initiate from Xbox Live Compute | ☑ |
| Can Accept on Xbox Live Compute | ☑ |
| Can Initiate from Windows Desktop | ☑ |
| Can Accept on Windows Desktop | ☑ |
| Can Initiate from Other Devices | ☑ |
| Can Accept on Other Devices | ☑ |
| 1 | 12 members |
| Name | WwiseCommand |
| Available in Shipping Builds | □ |
| Initiator Socket Description Name | WwiseCommandSocket |
| Acceptor Socket Description Name | WwiseCommandSocket |
| Can Initiate from Console | ☑ |
| Can Accept on Console | ☑ |
| Can Initiate from Xbox Live Compute | ☑ |
| Can Accept on Xbox Live Compute | ☑ |
| Can Initiate from Windows Desktop | ☑ |
| Can Accept on Windows Desktop | ☑ |
| Can Initiate from Other Devices | ☑ |
| Can Accept on Other Devices | ☑ |
| 2 | 12 members |
| Name | WwiseNotification |
| Available in Shipping Builds | □ |
| Initiator Socket Description Name | WwiseNotificationSocket |
| Acceptor Socket Description Name | WwiseNotificationSocket |
| Can Initiate from Console | ☑ |
| Can Accept on Console | ☑ |
| Can Initiate from Xbox Live Compute | ☑ |
| Can Accept on Xbox Live Compute | ☑ |
| Can Initiate from Windows Desktop | ☑ |
| Can Accept on Windows Desktop | ☑ |
| Can Initiate from Other Devices | ☑ |
| Can Accept on Other Devices | ☑ |

As a result, your `<GAME>/Config/XboxOne/XboxOneEngine.ini` should contain the following lines:

```
+SecureSocketDescriptions=(SecureSocketName="WwiseDiscoverySocket",bAvailableInShippingBuilds=False,SecureSocketProtocol=Udp,SecureSocketPortRange="24024",bCanInitiateConnections=True,bCanAcceptConnections=True,bCanSendChatData=False,bCanReceiveChatData=False,bCanSendGameData=False,bCanReceiveGameData=False,bCanSendDebugData=True,bCanReceiveDebugData=True,bCanSendInsecureData=False,bCanReceiveInsecureData=False)
+SecureSocketDescriptions=(SecureSocketName="WwiseCommandSocket",bAvailableInShippingBuilds=False,SecureSocketProtocol=Tcp,SecureSocketPortRange="24025",bCanInitiateConnections=True,bCanAcceptConnections=True,bCanSendChatData=False,bCanReceiveChatData=False,bCanSendGameData=False,bCanReceiveGameData=False,bCanSendDebugData=True,bCanReceiveDebugData=True,bCanSendInsecureData=False,bCanReceiveInsecureData=False)
+SecureSocketDescriptions=(SecureSocketName="WwiseNotificationSocket",bAvailableInShippingBuilds=False,SecureSocketProtocol=Tcp,SecureSocketPortRange="24026",bCanInitiateConnections=True,bCanAcceptConnections=True,bCanSendChatData=False,bCanReceiveChatData=False,bCanSendGameData=False,bCanReceiveGameData=False,bCanSendDebugData=True,bCanReceiveDebugData=True,bCanSendInsecureData=False,bCanReceiveInsecureData=False)
+SecureDeviceAssociations=(SecureDeviceAssociationName="WwiseDiscovery",bAvailableInShippingBuilds=False,InitiatorSocketDescriptionName="WwiseDiscoverySocket",AcceptorSocketDescriptionName="WwiseDiscoverySocket",bCanInitiateFromConsole=True,bCanAcceptOnConsole=True,bCanInitiateFromXboxLiveCompute=True,bCanAcceptOnXboxLiveCompute=True,bCanInitiateFromWindowsDesktop=True,bCanAcceptOnWindowsDesktop=True,bCanInitiate
FromOtherDevices=True,bCanAcceptOnOtherDevices=True)
+SecureDeviceAssociations=(SecureDeviceAssociationNa
me="WwiseCommand",bAvailableInShippingBuilds=False,I
nitiatorSocketDescriptionName="WwiseCommandSocket",A
cceptorSocketDescriptionName="WwiseCommandSocket",bC
anInitiateFromConsole=True,bCanAcceptOnConsole=True,
bCanInitiateFromXboxLiveCompute=True,bCanAcceptOnXbo
xLiveCompute=True,bCanInitiateFromWindowsDesktop=Tru
e,bCanAcceptOnWindowsDesktop=True,bCanInitiateFromOt
herDevices=True,bCanAcceptOnOtherDevices=True)
+SecureDeviceAssociations=(SecureDeviceAssociationNa
me="WwiseNotification",bAvailableInShippingBuilds=Fa
lse,InitiatorSocketDescriptionName="WwiseNotificatio
nSocket",AcceptorSocketDescriptionName="WwiseNotific
ationSocket",bCanInitiateFromConsole=True,bCanAccept
OnConsole=True,bCanInitiateFromXboxLiveCompute=True,
bCanAcceptOnXboxLiveCompute=True,bCanInitiateFromWin
dowsDesktop=True,bCanAcceptOnWindowsDesktop=True,bCa
nInitiateFromOtherDevices=True,bCanAcceptOnOtherDevi
ces=True)
```

For more information on how to modify the Xbox One AppxManifest, please refer to the Unreal Engine documentation.

**Q:** My game freezes when I deploy it to an iOS, XboxOne, or Switch device. Why?
**A:** The Unreal Engine built-in audio system needs to be disabled. To do so, please follow the appropriate instructions for your platform:

| Platform | Instruction |
|---|---|
| iOS | In `<UE4_ROOT>/Engine/Config/iOS/IOSEngine.ini`, replace<br>`AudioDeviceModuleName=IOSAudio`<br>with<br>`AudioDeviceModuleName=` |

| | |
|---|---|
| XboxOne | In `<UE4_ROOT>/Engine/Config/XboxOne/XboxOneEngine.ini`, replace<br>`AudioDeviceModuleName=XAudio2`<br>with<br>`AudioDeviceModuleName=` |
| Switch | In `<UE4_ROOT>/Engine/Config/switch/BaseSwitchEngine.ini`, replace<br>`AudioDeviceModuleName=SwitchAudio`<br>with<br>`AudioDeviceModuleName=` |

**Q:** My game crashes when I deploy it to a Switch device. Why?
**A:** The Unreal Engine built-in audio system needs to be disabled for Switch. To do so, please open the `<UE4_ROOT>/Engine/Config/switch/BaseSwitchEngine.ini` file, and replace the line

```
AudioDeviceModuleName=SwitchAudio
```

with

```
AudioDeviceModuleName=
```

**Q:** When I try to use the Wwise Convolution Reverb plug-in in Unreal Engine 4, why do I get the following error: `LogAkAudio:Error: Plug-in media unavailable: MediaID`?
**A:** For Unreal Engine 4.10 and earlier, plug-ins that require media (like the Impulse Response for Wwise Convolution Reverb) require that the SoundBank be edited in the Wwise application as documented on the AudioKinetic website. In Unreal Engine 4.11 and later, you can assign a SoundBank an Auxiliary Bus via the **Required Bank** field in the `UAkAuxBus` properties. This will add necessary plug-in media to the SoundBank.

**Q:** When running a packaged game, some or all of my sound are not

playing. Why?
**A:** The Wwise plug-in generates the SoundBanks in the WwiseAudio folder in your game's content folder. Since .bnk files are not recognized by Unreal, the WwiseAudio folder is added as an "Additional non-asset directory to package" in your game's packaging settings. For this reason, Unreal .uasset files (including AkAudioEvent and AkAudioBank) should **not** be placed in the WwiseAudio folder, as they will be incorrectly packaged twice.

# What's New?

Each version of this integration matches a specific build of Unreal Engine 4. Here is what has changed in each release of the integration (in addition to upgrading to the new Unreal build).

**Note:** This integration does not support experimental Unreal Engine 4 features.

### Migration Notes Known Issues

## Unreal Engine 4.15/4.16/4.17 - Wwise 2016.2.5.6121.484

- Removed support for UE 4.12, 4.13 and 4.14
- Added support for UE 4.17
- Since Unreal 4.17, the Unreal audio system needs to be disabled on the Xbox One and Switch platforms. Please refer to **Frequently Asked Questions** for more information.
- **WG-34388** Fixed: Crash in Editor when resetting RTPC sequence name to default in Level Sequencer.

## Unreal Engine 4.12/4.13/4.14/4.15/4.16 - Wwise 2016.2.5.6121.471

- **WG-32388** Fixed: `Max Attenuation` does not update Events that call Switches.
- **WG-32554** Fixed: Crash with `AkComponent::UpdateGameObjectPosition` when cooking Unreal project.

## Unreal Engine 4.12/4.13/4.14/4.15/4.16 - Wwise 2016.2.4.6098.451

- Added support for Unreal Engine 4.16
- **WG-31942** Fixed: Added support for Wwise file packages.
- **WG-33251** Fixed: Added missing include files when building non-Unity builds.

## Unreal Engine 4.12/4.13/4.14/4.15 - Wwise 2016.2.3.6077.435

- **WG-32914** Fixed: Avoid potential deadlocks and crashes in the AK Unreal IO code in a packaged game.

## Unreal Engine 4.12/4.13/4.14/4.15 - Wwise 2016.2.3.6077.422

- **WG-30020** Fixed: Allow possibility to dismiss the warning about missing Wwise project on editor startup.
- **WG-30695** Added support for the Switch platform.
- **WG-31076** Fixed: Marked all of the Wwise Blueprint nodes as `BlueprintCosmetic` to avoid running them on a dedicated server.
- **WG-31455** Added AkComponentCallbackManager. This allows better handling of required callbacks for an AkComponent, reducing concurrency risks.
- **WG-32046** Fixed: Add possibility to automatically start an AkAmbientSound on BeginPlay
- **WG-32490** Fixed: Deprecated level sequencer code
- **WG-32763** Fixed: Discontinued use of monolithic engine header files, like "Engine.h".
- **WG-32768** Fixed: Crash when loading large banks using EDL.
- **WG-32799** Fixed: Increased number of concurrent IO transfers in editor to speed up SoundBank loading.

## Unreal Engine 4.12/4.13/4.14/4.15 - Wwise 2016.2.2.6022.371

- **WG-32464** Allow opening an existing project in UE4.15 on the Mac platform.

## Unreal Engine 4.12/4.13/4.14/4.15 - Wwise 2016.2.2.6022.359

- **WG-31087** Added Level Sequence support.
- **WG-31687** Added Event Driven Loader support.
- **WG-31816** Added migration of Matinee tracks to Sequencer tracks.
- **WG-31924** Fixed: Editor crash when trying to post an Event on an AkComponent that is being auto-destroyed.
- **WG-32259** Fixed: Made AkComponent a Blueprintable component. Blueprint components can now use AkComponent as a base class.

## Unreal Engine 4.12/4.13/4.14 - Wwise 2016.2.1.5995.317

- **WG-31588** Fixed: Make sure no sound is played on a dying actor.
- **WG-31590** Improved low-memory conditions related to callbacks.
- **WG-31860** Fixed: Remove warnings found by clang static code analysis.
- **WG-31876** Fixed: Allow relative paths for the Wwise install path.
- **WG-31888** Fixed: Potential crash on exit due to accessing deleted FString.

## Unreal Engine 4.12/4.13/4.14 - Wwise 2016.2.0.5972.301

- **WG-31589** Fixed: Do not destroy a spawned component that is not set to auto-destroy and is failing to post an event.
- **WG-31678** Fixed: Streaming files can now be opened correctly.
- **WG-31771** Fixed: Avoid a crash when adding an AnimNotify_AkEvent to a playing animation.

## Unreal Engine 4.12/4.13 - Wwise 2016.2.0.5972.274

- **WG-29980** Language-specific folders are now parsed to retrieve max attenuation values for Events.
- **WG-30448** Improved logging of AkComponent when attempting to post an invalid AkEvent.
- **WG-30491** Fixed: Make sure to stop sounds when ending a PIE session.
- **WG-31030** Fixed: Optimized `FAkAudioDevice::Get()`.
- **WG-31040** Fixed: Memory leak in AkComponent when attempting to post an invalid AkEvent.
- **WG-31075** Fixed: Removed call to `FAkAudioDevice::Get()` from bank load callbacks in order to prevent crash in module manager.
- **WG-31186** Fixed: Prevent crash in `AkComponentCallback` by canceling Event callbacks when AkComponent gets destroyed.
- **WG-31204** Fixed: Memory leak when a spawned AkComponent failed to post its associated Event.
- **WG-31277** Fixed: Crash in editor when attempting to post an Event on a destroyed actor.

## Unreal Engine 4.11/4.12/4.13 - Wwise 2016.1.3

- **WG-30993** Fixed: Remove explicit LoadLibrary on XAudio 2.7.
- **WG-31015** Fixed: Link with the Recorder plug-in.

## Unreal Engine 4.11/4.12 - Wwise 2016.1.2

- **WG-30304** Fixed: "Unload stream level" no longer posts a global "Stop All" to the SoundEngine.
- **WG-30754** Fixed: `FAkAudioDevice::PostEvent` now always returns the playing ID.
- **WG-30804** Fixed: Removed dependencies on the Wwise SDK samples. The new IO system now uses Unreal IO utilities only.

## Unreal Engine 4.11/4.12 - Wwise 2016.1.1

- **WG-29972** Fixed: Multiple potential threading issues with the auto-destroy behavior of the AkComponent.
- **WG-29979** Fixed: EndOfEvent callbacks are now always called.
- **WG-30004** Fixed: SetGameObjectOutputBusVolume is now exposed in Blueprints and AkAudioDevice.
- **WG-30404** Fixed: The attenuation scaling factor now properly works on `AkComponents`.
- **WG-30409** Fixed: It is now possible to decode Vorbis-encoded files.

## Unreal Engine 4.12 - Wwise 2016.1

- **WG-29991** Added `AkEvent String Input Field` to `AkAmbientSound`.
- **WG-29997** Suppressed duplicate "LogAkAudio: StopAll API called" entries in the output log.
- **WG-30218** Fixed: Crash when connecting the Wwise Profiler to the Android platform.
- **WG-30255** Fixed: UE4 crash when adding key to `Ak Event Track` in Matinee.

- **WG-29991** Added `AkEvent String Input Field` to `AkAmbientSound`.
- **WG-29997** Suppressed duplicate "LogAkAudio: StopAll API called" entries in the output log.

## Unreal Engine 4.11.2 - Wwise 2016.1

- **WG-29917** Fixed: Fixed a case where the attenuation radius of `AkAmbientSounds` did not show up in the Editor window.
- **WG-30000** Fixed: Changed meta properties of `StartAllAmbientSounds` and `StopAllAmbientSounds`.
- **WG-30012** Fixed: Integrated GitHub pull request #5. Show project supported platforms as "Available Platforms" in the "Generate Sound Banks" window.
- **WG-30014** Fixed: Fixed a crash when starting a Play in Editor session that uses a dedicated server.
- **WG-30031** Fixed: Removed usage of the `World` global pointer.
- **WG-30205** Fixed: On the Mac platform, the Wwise.app path may now contain spaces.

## Unreal Engine 4.11 - Wwise 2015.1.7

- **UI-316** Fixed: Prevent a crash when starting a Play in Editor session with a dedicated server.
- **WG-29917** Fixed: Make sure the attenuation radius is visible in the Editor.
- **WG-29944** Fixed: Stop ambient sounds when stopping PIE session.

# Unreal Engine 4.11 - Wwise 2015.1.6

- The Wwise Unreal integration is now a plug-in. Please see **Migrating from the UE4 Wwise integration source code to the plug-in version** for migration notes.
- **UI-273** Fixed: Create a new uasset type for Auxiliary Busses. This allows to assign an AuxBus to a SoundBank. See **AkAuxBus** for more information.
- **UI-280** Fixed: Deprecated "...by name" methods, and replaced then with a string field in the relevant methods. See **Migrating "...by name" methods** for more information.
- **UI-309** Fixed: Made the Wwise project path relative to the game folder, instead of relative to the UE4Editor.exe file. See **Migrating the Wwise project path** for more information.

## Unreal Engine 4.10 - Wwise v2015.1.4

- **UI-265** Fixed: In the AnimNotifies, test the AkComponent's validity before using it. This reduces log spam when using UE with the `-nosound` option.
- **UI-270** Fixed: It is now possible to build in non-Unity mode.

## Unreal Engine 4.9 - Wwise v2015.1.2

- **UI-249** Fixed: Avoid crashing when no Audio Playback device is available on Windows.
- Added a "Spawn AkComponent at location" Blueprint node. See **Spawn Ak Component at Location** for more information.
- Added support for the Mac Editor.
- Added support for the Mac platform.
- Added support for the iOS platform.
- Added support for the Linux platform.

## Unreal Engine 4.8 - Wwise v2015.1

- Migrated the UE4.8 integration to Wwise 2015.1
- Fixed a crash on Xbox One if the app manifest did not contain a definition for Wwise communication ports.

## Unreal Engine 4.8 - Wwise v2014.1.5

- **UI-206** Fixed: Sounds with an attenuation can now be heard when an animation window is opened.
- **UI-212** Fixed: Added a "Load Init Bank" Blueprint node.
- **UI-213** Fixed: Added null checks in Blueprint nodes, preventing crashes.
- **UI-214** Fixed: No crash when saving an output capture to disk.
- **UI-215** Fixed: No issue with the Attenuation Scaling Factor of an AkComponent not being set when attached to an actor.
- **UI-217** Fixed: Made AkReverbVolume's collision settings visible in the details panel.
- **UI-220** Fixed: Fixed the occlusion fade behavior for sounds that a spawn occluded.
- **UI-223** Fixed: The gathering of the max attenuations for AkEvent does not halt when the parsing of one SoundBank fails.
- **UI-226** Fixed: SoundBank generation is now performed by the 32-bit WwiseCLI.exe when the 64-bit WwiseCLI.exe is missing.
- **UI-230** Fixed: Now using `OnComponentDestroyed` instead of `FinishDestroy` to unregister Wwise Game Objects, which had caused some Game Objects to be registered for too long.
- **UI-233** Fixed: Added validity checks on actors in the GetGameObjectID function.
- **UI-234** Fixed: Multiple viewports in the SetListener function are properly handled. This fixes a crash in AkAudioDevice.cpp.
- **UI-236** Fixed: The Location Type given to the GetAkComponent Blueprint node is properly handled.
- **UI-239** Fixed: No crash at engine shutdown as a result of the Bank Manager unloading SoundBanks that might already have been destroyed.
- Added a "Follow" check box to the AnimNotify_AkEvent. Leaving it unchecked will post the AkEvent at a specific location, instead of attaching to the parent.
- Added a new AnimNotify: AnimNotify_AkEventByName, allowing to post events using their name as a string.
- Added Debugging tools to Blueprints. For more information, see **Debug Blueprint Functions**

## Unreal Engine 4.7 - Wwise v2014.1.3

- Added support for the Android platform.
- Now including a new demonstration game, based on Epic's First Person Template. For more information, see **Sample Game**.
- All Wwise Integration settings now have their page in the Unreal Project Settings window. No need to manually edit INI files anymore!
- **UI-157** Fixed: When generating the SoundBanks, add the Max Attenuation information to AkEvents. It can be retrieved via Blueprints.
- **UI-187** Fixed: Allow the preview of AkEvents from the Content Browser by pressing the space bar.
- **UI-188** Fixed: When an AkAmbientSound is about to be destroyed, cancel its event callback to avoid using a destroyed object in the callback.
- **UI-189** Fixed: Added an attenuation scaling factor to AkComponent, allowing to make each actor's attenuation radius unique.
- **UI-190** Fixed: Added a new Blueprint node that allows setting the exact list of loaded SoundBanks. For more information, see **Load Banks**
- **UI-193** Fixed: Get the `UAkAudioEventFactory` directly when creating `UAkAudioEvents` by Drag & Drop.
- **UI-194** Fixed: Optimized the finding of `AkReverbVolumes` at a location.
- **UI-195** Fixed: Added **blueprint_actor_posteventbyname**, **features_blueprintcsoundbanks_loadbyname**, and **Unload Bank** Blueprint nodes.
- **UI-196** Fixed: Added a configuration parameter for the maximum number of concurrent Reverb Volumes applied on a sound.
- **UI-201** Fixed: Added an UAkAudioEvent to AkComponent, to bring it in sync with Unreal's AudioComponent. An AkComponent can by added in a Blueprint via drag & drop of a UAkAudioEvent. An actor with an attached AkComponent will also show its attenuation radius in the Editor.
- **UI-204** Fixed: Added code to unload auto-loaded SoundBanks.
- **UI-205** Fixed: Don't post events in a world that does not allow audio playback.
- **UI-208** Fixed: Fixed a potential crash when generating SoundBanks.

## Unreal Engine 4.6 - Wwise v2014.1.1

- **UI-183** Fixed: Use the debug SoundEngine libraries when UnrealBuildTool is configured to use the debug CRT libraries (`bDebugBuildsActuallyUseDebugCRT = true`).
- **UI-184** Fixed: Added a billboard component to AkComponent to make them visible in the Editor.
- **UI-186** Fixed: Allow drag & drop of .bnk files in the Content Browser to create UAkAudioBank assets.

## Unreal Engine 4.5 - Wwise v2014.1

- **UI-172** Fixed: Resolved assert created by the AkReverbVolumeChannel collision channel.
- **UI-174** Fixed: Optimized the "Generate SoundBanks" window.
- **UI-175** Fixed: Added preprocessor guards to remove compilation errors on platforms not supported by this integration.

## August 2014 - Wwise v2014.1

- **UI-166** Fixed: Auxiliary Bus properties are now applied on Ambient Sounds contained within Reverb Volumes.
- Updated Wwise version to 2014.1. If you wish to keep on using 2013.2.x, see **Frequently Asked Questions**.

## August 2014 - Wwise v2013.2.9

- Added Japanese documentation.

## July 2014 - Wwise v2013.2.9

- Added a SwitchDemo map to the UnrealWwiseDemo sample game, demonstrating the use of switches.
- **UI-161** Fixed: Removed the duplicate AkComponent created when starting an ambient sound. This prevented the Stop Ambient Sound method from working.

## June 2014 - Wwise v2013.2.8

- **UI-152** Fixed: Add SoundBanks and streamed files to the packaging process.

## April 2014 - Wwise v2013.2.7

- **UI-149** Fixed: Set proper default values for AkReverbVolumes.

## March 2014 - Wwise v2013.2.6

- Visual Studio 2013 is now used by default. Please refer to the **Requirements** page for more details.
- **UI-146** Fixed: Fixed non-unity build compilation errors.
- **UI-147** Fixed:
  - Verify the SoundEngine is initialized in the AkComponent tick function;
  - Properly set the collision channel name for AkReverbVolumes;
  - Avoid loading banks in a commandlet;
  - Removed the automatic bank creation mechanism.
- **UI-148** Fixed: Use "Get Player Controller" in the RTPCDemo Blueprint to ensure sound keeps playing when transitioning between "Simulate" and "Possess" modes in Editor.

## January 2014 - Wwise v2013.2.5

- Added an example of AkAnimNotify in ShooterGame. It is located in the FPP_RifleReload animation.
- **UI-131** Fixed: Spatialized sounds are now audible in the Animation Editor: Added a second listener for the Animation Editor window, and routed game objects created in this window to the new listener. **WARNING:** This fix changes the AnimNotify_AkEvent Blueprint. If you have made modifications to it, be sure to keep a backup before merging this integration.
- **UI-134** Fixed: Removed the input flag StopWhenOwnerDestroyed to GetAkComponent. It was unused and could create confusion. Added a SetStopWhenOwnerDestroyed method to the AkComponent.
- **UI-136** Fixed: Allow Wwise Authoring to communicate with an Xbox One application.
- **UI-137** Fixed: Make sure a temporary game object (created by "Post Event at Location") is subject to AAkReverbVolumes.
- **UI-138** Fixed: Prevent the Sound Engine from creating UE errors while running in a commandlet. This is a workaround. Maybe the Sound Engine should not be initialized at all.
- **UI-139** Fixed: Removed FMath::Abs on Z projection when setting listener position in SoundEngine (Unreal code has been left untouched). This allows upside down listeners.
- **UI-145** Fixed: Fixed an error in the AkComponent::SetRTPCValue method that prevented RTPCs from being properly applied.

## December 2013 - Wwise v2013.2.4

- Unreal Wwise integration now officially supports PS4 and Xbox One platforms!
- Added platform selection to bank generation dialog box.
- Added bank generation dialog box to build menu.
- `LinkedProject` configuration parameter has been moved from `Engine/Config/BaseEditor.ini` to `<Your Game>/Config/DefaultGame.ini`.
- Added simple occlusion support.
- Fixed a crash when getting the UAkComponent from another component which has no owner.
- **UI-130** Fixed: Fixed performance issues with the UAkComponent::UpdateAkReverbVolumeList method.
- **UI-130** Fixed: Added a flag to the UAkComponent specifying if it is influenced by reverb volumes.
- **UI-132** Fixed: Prevent UAkComponents from ticking on server configurations.
- **UI-135** Fixed: Expose RTPC interpolation time.

# October 2013 - Wwise v2013.2.1

- Added bank generation for PlayStation4 and Xbox One.
- Added AkReverbVolume, allowing for mapping a volume in space to a Wwise auxiliary bus.
- **UI-128** Fixed: Implemented asynchronous IO access.

## September 2013 - Wwise v2013.2.1

- Fixed build settings for Visual Studio 2012.
- Fixed UEngine::UseSound() returning false when only AkAudioDevice is present.
- Fixed crash in UAkComponent::PostAkEvent with null event.
- Fixed game object position with static actor.

## August 2013 - Wwise v2013.2

- Fixed confusion between Actor, AkComponent, the Wwise game object and their respective lifespans:
  - AkComponent follows attachment semantics of Epic's AudioComponent. AkComponents are re-used when all attachment parameters match.
  - AkGameplayStatics contains global helpers targetting Actors for easy use in blueprints.
- Fixed missing Events and Banks in SoundBank Definition File.
- Fixed game object names in Wwise Profiler.
- Generating SoundBanks now triggers a refresh of loaded banks.
- Set RTPC Value global helper now supports setting a Game Parameter at global scope (no Actor target).

## July 2013 - Wwise v2013.1.1

- Added message box in case of error during SoundBank Generation.
- New, distinct colors for Audiokinetic Event and Audiokinetic Bank asset types in the Content Browser.
- Fixed compilation on Xbox One and PS4 platforms.
- Fixed issue with AnimNotify_AkEvent using the wrong type.
- Fixed update of AkComponent Game Object position.

# June 2013 - Wwise v2013.1.1

- Added DebugGame build configuration support.
- Partial Wwise SDK now included to help setting up the integration.
- Fixed xaudio2/mmdev DLL unload issues in Windows 64-bit build.
- Stopped matinee movie from playing in demomap - was preventing user input from working.
- Fixed AkAudioDevice listener position update in FLevelEditorViewportClient::UpdateAudioListener.
- Changed the AudioDeviceModuleName default value to XAudio2 on the windows platform.

## May 2013 - Wwise v2013.1.1

- Introduced doxygen documentation.
- Now building using Wwise 2013.1.1 build 4677.

## March 2013 - Wwise v2013.1

- Now building using Wwise 2013.1 beta build 4609.

# Wwise Unreal Plug-in » What's New?

# Migration Notes

## Migrating to the UE4.11/4.12 Wwise 2016.1.1 integration.

As part of the fix for WG-30404, the `AttenuationScalingFactor` `UPROPERTY` of `AkComponent` has been made read-only in Blueprints. To set its value, you now need to call the `SetAttenuationScalingFactor` method on `AkComponent`.

# Migrating from the UE4 Wwise integration source code to the plug-in version

Starting from Unreal Engine 4.11, the Wwise UE4 integration will now be distributed as a plug-in.

**NOTE:** Events can no longer be dragged from Wwise and dropped in the Unreal Content Browser to create corresponding AkAudioEvent objects. They can now be dragged directly from the Wwise Picker into the Unreal Content Browser to create their corresponding objects.

To update a UE4 project from UE 4.10 (or older) to UE 4.11 (or newer):

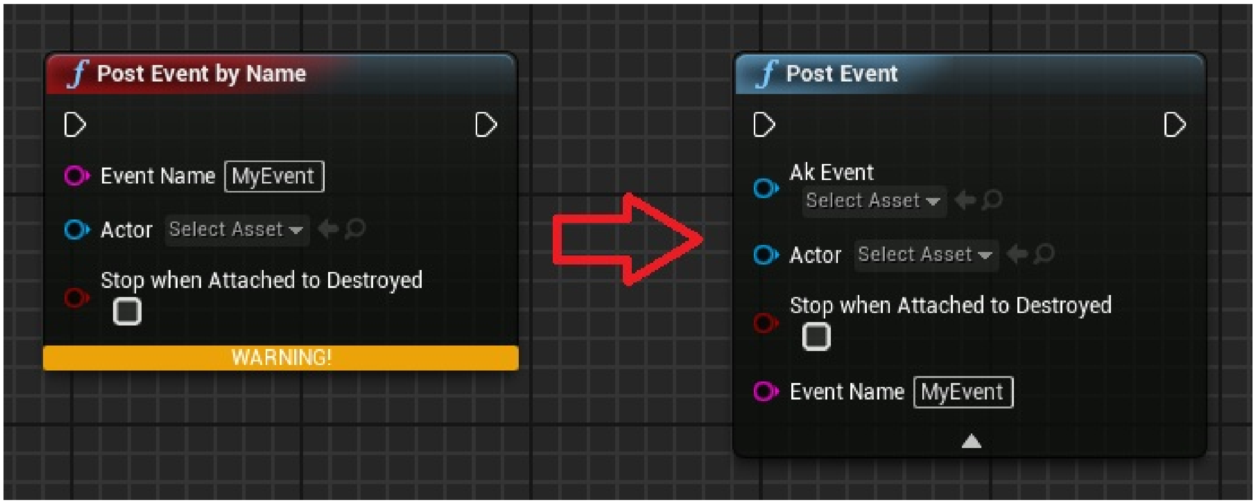
1. Back up your game project.
2. Install Unreal Engine from the Epic Games Launcher or compile your own Unreal Engine from source code.
3. Install the Wwise Unreal plug-in as an Engine or a Game plug-in, as outlined in **Installation**. Make sure that you use the SDK files you built when doing so.
4. Load your game project in the Unreal Editor.
5. If you installed the Wwise Unreal plug-in as an Engine plug-in, enable the Wwise plug-in by going to **Edit > Plugins > Audio** and selecting the "enable" option under the Wwise Unreal Engine 4 Integration section. You will need to reload your project to enable the plug-in.
6. If your project uses the AkEvent or the AkEventByName AnimNotify, you will be prompted to reload your project as some assets were modified to support the new plug-in model.

The last step is necessary because the AnimNotifies provided with the plug-in had to be moved from the Engine content folder to the plug-in's own content folder. Moving the references requires a project reload.

Once all the steps have been completed, you are ready to use the Wwise Unreal plug-in.

# Migrating "...by name" methods

Starting with the 4.11 version of the Wwise Unreal plug-in, all "...by Name" methods (for example, PostEventByName or LoadBankByName) are deprecated. It is still possible to use strings to post Events or load SoundBanks by using the advanced fields in the Blueprint nodes. For example, if you wish to post an Event by name, you should now use the "Event Name" field of the "PostEvent" Blueprint node:



Post Event by Name: Event Name MyEvent

Post Event: Event Name MyEvent

# Migrating the Wwise project path

Starting with the 4.11 version of the Wwise Unreal plug-in, the Wwise project path, found in the Wwise settings, is now relative to the game folder, instead of the UE4Editor.exe file. The change should be automatically done the first time you run the 4.11 Wwise Unreal plug-in.

# Wwise Unreal Plug-in » What's New?

# Known Issues

- **UI-151** Use Reverb Volumes flag doesn't work in real time: The Blueprint box "Set use reverb volumes" does not work as expected.
- **UI-277** It is not possible to connect the Wwise Profiler to a Linux game.
- **UI-307** Freeze on iOS platform. Make sure to disable the Unreal Engine audio device. See **Frequently Asked Questions** for more information.
- **WG-32685** It is not possible to connect the Wwise Profiler to a Switch game.
- Freeze on Switch platform. Make sure to disable the Unreal Engine audio device. See **Frequently Asked Questions** for more information.