

All actions, events and parameters from your game are communicated from the Audio Translation Layer (ATL) to the selected audio middleware using Audio System Controls which are mapped to one (or several) controls inside your middleware. The connection between the Audio System Controls and the audio middleware, as well as the creation of the Audio System Controls themselves, are achieved with the Audio Controls Editor (ACE) which is explained in detail in this article.



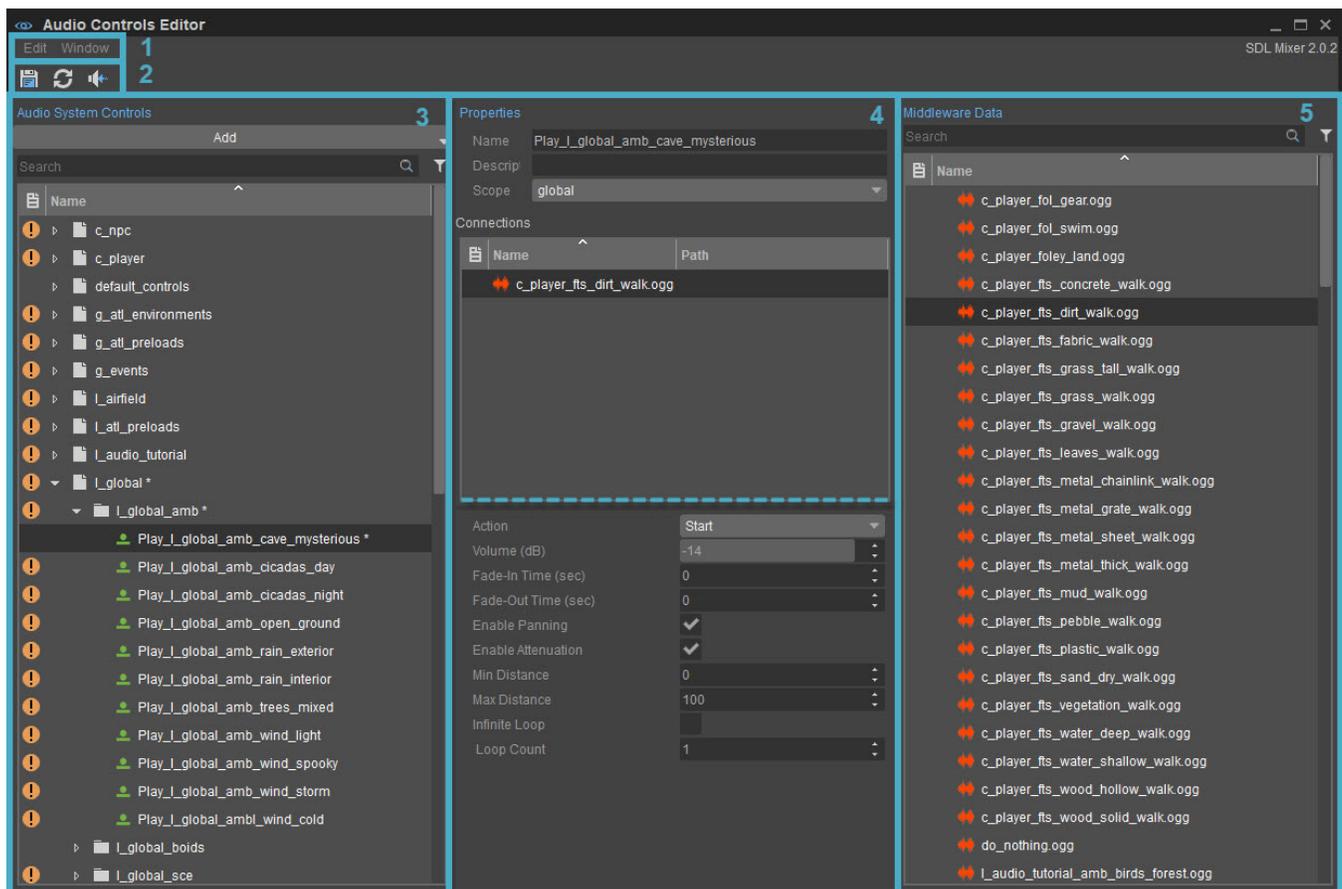
In release 5.5, the ACE underwent some big changes. If you started your project before this release and updated the engine to 5.5, please read the Migration Guide at the bottom of this page.

You can find the Audio Controls Editor inside the Sandbox Editor by going to the menu found in **Tools -> Audio Controls Editor**.

Depending on the audio middleware you are using, the appearance of the Audio Controls Editor changes slightly. However, it always comprises of the three main panels - the Audio System Controls, the Properties panel and the Audio Middleware Data panel.



These panels can be docked and undocked, so you can change the layout to suit your preferences.



As you can see, icons are sometimes used in the Audio Controls Editor. If you hover over them, a tooltip will be displayed telling you what the problem is.

These icons will not only just be displayed in front of a control, but also in front of its parent folders and libraries when these are collapsed, to let you know that there is a problem with a control somewhere in that folder/library.

The data for the ACE is loaded automatically when the tool is opened. To save any changes made to the controls click the **Save All** button. To reload the data from disk click the **Reload All** button.

- 1. Menu Bar
 - Edit
 - Window
- 2. Toolbar
- 3. Audio System Controls
 - Control Types
 - Search Bar

- Filters
- Context Menu
- Adding Controls/Folders/Libraries/States
- Moving Controls
- Editing Controls
- Deleting Controls
- Previewing Triggers
- 4. Properties Panel
 - Properties
 - Connections
 - Context Menu
 - Connecting Controls
- 5. Middleware Data Panel
 - Overview
 - Filtering
 - Connecting the Middleware Controls to the System Controls
 - Context Menu
- Resource Selectors
 - Search Bar
 - Context Menu
- Advanced Users
 - Data Handling
- Changes in Release 5.5
 - Migration Guide
 - CVars

1. Menu Bar

Edit

Option	Description
Preferences	<p>If the selected audio middleware support projects, the project path can be edited here.</p> <p>The default path for Wwise is: <code>audio/wwise_project</code>.</p> <p>The default path for Fmod Studio is: <code>audio/fmod_project</code>.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p> As SDLMixer doesn't work with projects, you can't change anything here if SDLMixer is used.</p> </div>

Window

Option	Description
Add Pane	Lets you add the three panes (Audio System Controls, Middleware Data, Properties) in case they were closed.
Reset layout	Resets the layout of the ACE to the default layout.
Audio System Controls	If the Audio System Controls panel is closed, you can open it again with this option.
Middleware Data	If the Middleware Data panel is closed, you can open it again with this option.
Properties	If the Properties panel is closed, you can open it again with this option.

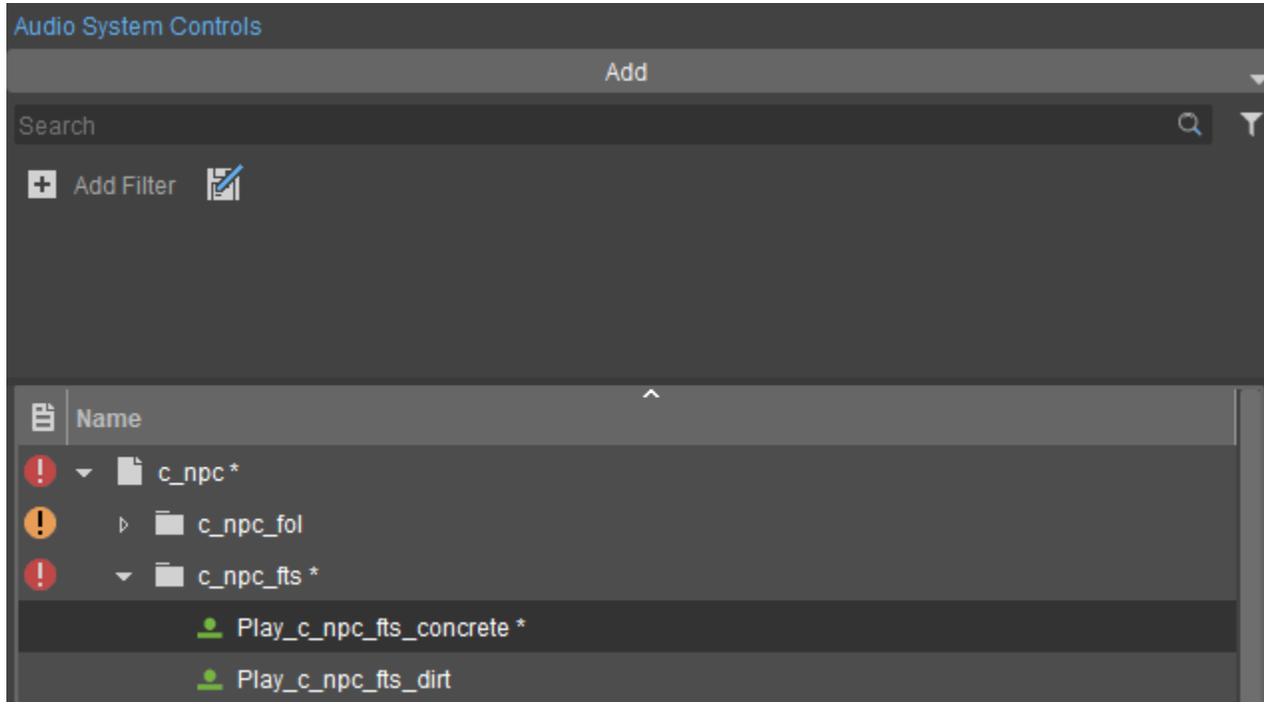
2. Toolbar

Button	Name	Description
	Save All Changes	Lets you save any changes made in the ACE.
	Reload all ACE and middleware files	Reloads all ACE and middleware files. Basically undoes all changes up to the last time you saved your changes.
	Refresh Audio System	Reloads all audio system data (soundbanks and audio files).

3. Audio System Controls

The Audio System Controls panel shows all the controls in your CRYENGINE project. It allows you to create, move and delete the 5 different types of controls available ([Triggers](#), [Parameters](#), [Switches](#), [Environments](#) and [Preload Requests](#)). You can also group your controls in folders which can be useful when organizing large projects.

Whenever a change has been made to a Trigger, Switch, etc., the item and all the folders above will be marked with an asterisk so you know where the changes have been made:



Control Types

Control Type	Icon	Description
Triggers		Triggers are containers that execute the audio middleware events connected to them. A Trigger might have any number of Events connected to it.
Parameters		RTPCs (Real-Time-Parameter-Controls) are used to seamlessly alter a parameter's value on which the Audio Middleware can receive and drive corresponding effects.
Switches		A Switch is a collection of States, each of which manipulate the connected parameters of a middleware in a certain way. An example of a commonly used Switch would be that for character footsteps, that plays different footstep sounds depending on the surface (concrete, grass, etc.).
Environments		Environments are used to drive effects like reverbs and weapon reflections, in areas such as AreaBox, AreaShape, AreaSphere or AreaSolid that are found under the Area option of the Create Object tool.
Preloads		Preloads are requests to load audio specific files (such as SoundBanks in an FMOD Studio implementation) into memory. Any number of files can be grouped into a single Preload request.
Folders		Folders are only used for structuring the Audio System Controls in the ACE, and have no functionality associated with them.

Search Bar

Typing anything in this Search bar will show you only audio controls that contain the letters that you typed.

Filters

You can filter the list of controls by using the **Filter** button and toggling the visibility of the control types you want to see.



The Invert button () inverts the filter, so show everything *except* what you selected.

Context Menu

When right clicking on an item in the Audio System Controls (or an empty space), you are presented with a combination of the following options, depending on what you clicked on:

Option	Description
Load Global Preload Request	NOTE: Only appears when the Scope of the selected Preload Request is Global and the Auto Load box is unticked. Manually loads all soundbanks connected to the preload request.
Unload Global Preload Request	NOTE: Only appears when the Scope of the selected Preload Request is Global and the Auto Load box is unticked. Manually unloads all soundbanks connected to the preload request.
Execute Trigger	Executes the Trigger you right clicked on (only available when clicking on Triggers). NOTE: You can also execute a trigger by double clicking it or pressing the space bar.
Add	Adds either a library or a folder or an Audio System Control, depending on the current selection. A library will always be created on the root level Everything else will be added as child of the current selection. Except for a parent folder, which will be created as a child of the currently selected parent.
Open Containing Folder	Opens the folder that contains the selected control in the Windows Explorer. NOTE: Only appears for actual files.
Rename	Renames the control. Note: Default controls cannot get renamed.
Delete	Deletes the control. Note: Default controls cannot get deleted.
Expand Selection	Expands the selected folder.
Collapse Selection	Collapses the selected folder.
Expand All	Expands all folders.
Collapse All	Collapse all folders.

Adding Controls/Folders/Libraries/States

To add new controls, folders, libraries etc., press the **Add** button and select the type of item you want to add. This will add a new control or folder as a child of the currently selected item in the list. If none is selected it will be created at the root level.

 The **Add** menu will only show menu items that can be as a child of the currently selected control, switch, etc.

You can also right click on any control or folder to access the same options with a context menu.

 When you're using FMod Studio or Wwise, States can only be added to Switches. To do this, select the Switch you want to add it to and either click the **Add** button and choose **State**, or right click on the Switch you want to add it to and choose **Add -> State**.
NOTE: You can also create a state by dropping a middleware parameter or VCA onto a switch.
NOTE: Currently also only parameters, switches and preload requests can be added when using Fmod Studio or Wwise.

Moving Controls

To move controls or folders between other items in the list just *drag and drop*. You can also *drag and drop* entire folders which will move their content along with it.

 Folders and controls cannot be moved to the root level. Items in the list cannot be moved into or out of of the default_controls library.

Editing Controls

In the ATL Controls panel you can change the name of any control or folder by pressing **F2** on your keyboard with the control selected. For further editing please look at the Properties panel. The same functionality can be accessed by right-clicking the control that is required to be modified.

Deleting Controls

You can delete controls by pressing the **Delete** key on your keyboard. The same functionality can be accessed by right clicking the control that wants to be deleted.



NOTE:

An Audio System Control name can only exist once per control type and a State name can exist once per Switch. For example, if you named an Audio Trigger "my_trigger1", there cannot be another Audio Trigger anywhere in your project with the same name. You will notice that as the ACE will automatically prevent creation of another Audio Trigger with the same name by renaming the control to "my_trigger2", "my_trigger3" and so on.



ATL Default Controls

The Audio System houses some default Audio Controls which it auto-creates if they are missing.

These controls are hard coded as they are generic and apply to any and every project regardless of its nature. Make sure to connect these controls to corresponding representations of your audio middleware to establish a functioning production environment.

All default controls are stored in the default_controls library. They cannot get deleted, renamed or moved out of that library. Also other controls cannot get moved into the default_controls library.

Audio Trigger

- **get_focus** - is sent if the application window gets focus. Automatically unmutes all sounds in Fmod Studio and SDL Mixer. In Wwise it needs to be connected to an unmute all event.
- **lose_focus** - is sent if the application window loses focus. Automatically mutes all sounds in Fmod Studio and SDL Mixer. In Wwise it needs to be connected to a mute all event.
- **mute_all** - is sent by the Sandbox's Mute Audio button if not already muted. Automatically mutes all sounds in Fmod Studio and SDL Mixer. In Wwise it needs to be connected to a mute all event.
- **unmute_all** - is sent by the Sandbox's Mute Audio button if mute_all was sent beforehand. Automatically unmutes all sounds in Fmod Studio and SDL Mixer. In Wwise it needs to be connected to an unmute all event.
- **pause_all** - Automatically pauses all sounds in Fmod Studio and SDL Mixer. In Wwise it needs to be connected to a pause all event.
- **resume_all** - Automatically resumes all paused sounds in Fmod Studio and SDL Mixer. In Wwise it needs to be connected to a resume all event.
- **do_nothing** - does nothing, is used to override the automatic stopping of StartTrigger instances by being applied to the StopTrigger property. Not displayed in the ACE, only in the resource selector, because it should not be edited.
Note: Connecting any of the above triggers in the ACE will override their default behavior.

Audio Parameters

- **absolute_velocity** - is updated on Audio Objects that have object_velocity_tracking switch enabled.
- **relative_velocity** - is updated according to the movement speed of the respective audio object in the level.

Audio Switches

- **occlusion_calculation_type** - is used to determine whether an Audio Object calculates obstruction and occlusion values.
- **absolute_velocity_tracking** - is used to determine whether an Audio Object should calculate its velocity when moving and updating the object_speed parameter.
- **relative_velocity_tracking** - is used to determine whether an Audio Object should calculate movement speed and updating the object_doppler parameter.
NOTE: The switches are not displayed in the ACE, only in the resource selector, because they should not be edited.

Previewing Triggers

It is possible to preview an Audio Trigger in three ways:

1. **Right-click** the trigger to open its context menu and choose **Execute Trigger**.
2. Press the **Spacebar** on the selected Audio Trigger.
3. **Double-click** the selected Audio Trigger.



1. Make sure to save changes made to Audio Triggers to be previewed, otherwise those changes won't be audible.
2. Executed triggers will stop once deselected.

4. Properties Panel

The Properties panel shows and allows you to edit all the properties of the control currently selected in the Audio System Controls panel including making connections to any of the matching audio middleware controls.

The properties displayed in the top half of the Properties panel are:

Properties

Property	Description
Name	The name of the control. This can also be edited in the Audio System Controls panel.
Description	Here you can type a short description of the Audio System Control. This will also appear on mouse-over in the Resource Selector (see further down on this page for more information about Resource Selectors).
Scope	<p>Controls can exist in a <i>Global</i> or on a <i>per-level</i> scope. A control with a <i>Global</i> scope will exist as long as your game is running and regardless of whether the control is used or not in the current level. When a specific level is defined as the scope they will only exist when that level is loaded. This setting is very useful in low memory systems, because controls are only loaded in levels where they are needed.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p> When a Default Control is selected, its Scope is forced to Global. Since this is the only setting for Default Controls, the Scope property is hidden when a Default Control is selected.</p> <p>The scope can't be set on states; they will inherit the scope of their parent switch.</p> </div>
Auto Load	NOTE: Only available for preload requests. Automatically loads all connected soundbanks when enabled. If the Scope is Global , the soundbanks are loaded on game start. If the scope is a level, the soundbanks are loaded on level start.

Connections

This is where you see the middleware controls that are connected to your audio system controls. See below under **Connection Controls** how to do this.

When you select a connected middleware control under **Connections** in the **Properties** panel, different properties appear in the bottom half of the Properties panel, depending on which audio middleware you're using, what kind of control you select and what kind of file you connect it with.

SDL Mixer

When connecting an **Audio Trigger** to an **audio file**, the following options are available:

Option	Description
Action	<ul style="list-style-type: none"> • Start - Indicates that the audio file will start playing when the trigger is executed. • Stop - Indicates that the audio file will be stopped when the trigger is executed. If the audio file was not playing or had already stopped the event will be ignored. • Pause - Indicates that the SDL Mixer event will be paused when the trigger is executed. • Resume - Indicates that the SDL Mixer event will be resumed when the trigger is executed.
Volume (dB)	Maximum volume at which the audio will play when not attenuated.
Fade-In Time (sec)	Fade-in time in seconds.
Fade-Out Time (sec)	Fade-out time in seconds. Used when the event is stopped.
Enable Panning	Allows the playing of the audio file to be 3D positioned so that it will be panned between speakers depending on the direction the player is looking at.

Enable Attenuation

When ticking this box, the following options will appear:

Option	Description
Min Distance	Distance at which the sound will start to attenuate.
Max Distance	Distance at which the sound will be completely attenuated and inaudible.

Looping

Option	Description
Infinite Loop	Loops the sound infinitely when you're within the maximum distance that the sound is played.
Loop Count	If the Infinite box was not ticked, this option will appear. This is the number of times the sound is played when the trigger is executed.

When connecting an **Audio System Controls parameter** to an **audio file**, the following options are available:

Option	Description
Multiply	The audio parameter value will be multiplied by this before being used to control the events volume. The value cannot be lower than 0 .
Shift	The audio parameter value will be incremented by this before being used to control the events volume. The values are clamped between -1 and 1 . Note: If using multiply and shift, shift is applied to the value before the multiplier.

When connecting a **Audio System Controls State** to an **audio file**, the following options are available:

Option	Description
Volume (normalized)	The connected event will get set to this volume. It's clamped between 0 and 1 .



When using the SDL Mixer implementation, parameters and switches/states can only control the volume of audio events. The value of parameters and states are normalized volumes (0 - 1).
0 = silent (-96dB)
1 = the volume which is set in the event connection of a trigger with a start action. Setting a parameter higher than **1** will not increase the volume beyond the value which is set in the trigger connection.
Parameters and switches/states are set per object and affect all triggers that use the same asset on that object.

Wwise

When connecting an **Audio System Controls parameter** to a **Wwise parameter**, the following options are available:

Option	Description
Multiply	The audio parameter value will be multiplied by this before being assigned to the Wwise parameter.
Shift	The audio parameter value will be incremented by this before being assigned to the Wwise parameter. Note: If using multiply and Shift , Shift is applied to the value before the multiplier.

When connecting an **Audio System Controls State** to a **Wwise parameter**, the following options are available:

Option	Description
Value	Value to set to the Wwise parameter when the state is activated.

When connecting an **Audio System Controls Preload Request** to a **Wwise soundbank**, the following options are available:

Option	Description
<platform name>	If enabled, the connected soundbank gets loaded when the preload request is called on this platform.

Fmod Studio

When connecting a **State** to an **Fmod Parameter** or **Fmod VCA**, the following options are available:

Option	Description
--------	-------------

Value	Value to set to the Fmod Event or Fmod VCA when the state is activated.
--------------	---

When connecting an audio parameter to an Fmod Parameter or Fmod VCA, the following options are available:

Option	Description
Multiply	The audio parameter value will be incremented by this before being assigned to the Fmod Parameter or VCA.
Shift	The audio parameter value will be incremented by this before being assigned to the Fmod parameter or VCA. NOTE: If using multiply and shift, shift is applied to the value before the multiplier.

When connecting an **Audio Trigger** to an **Fmod Event**, the following options are available:

Option	Description
Action	<ul style="list-style-type: none"> • Start - Indicates that the Fmod event/snapshot will start playing when the trigger is executed. • Stop - Indicates that the Fmod event/snapshot will be stopped when the trigger is executed. • Pause - Indicates that the Fmod event/snapshot will be paused when the trigger is executed. • Resume - Indicates that the Fmod event/snapshot will be resumed when the trigger is executed.

When connecting an Audio Trigger to an Fmod Snapshot, the following options are available:

Option	Description
Action	<ul style="list-style-type: none"> • Start - Indicates that the Fmod snapshot will start playing when the trigger is executed. • Stop - Indicates that the Fmod snapshot will be stopped when the trigger is executed.

When connecting a Audio System Controls Preload Request to an Fmod bank, the following options are available:

Option	Description
<platform name>	If enabled, the connected bank is loaded when the preload request is called on this platform.

Context Menu

When right clicking on a connection in the Properties panel, you get the following options:

Option	Description
Remove Connection	Disconnects this Middleware Control from the Audio Systems Control.
Select in Middleware Data	Selects the middleware control in the Middleware Data panel.

Connecting Controls

To create new connections between audio system controls and audio middleware controls just drag the control from the Audio Middleware Controls panel into the Connections area. An audio middleware control can also be dragged directly into the Audio System Controls panel, this will create a new control with the name of the audio middleware control and automatically connect both of them.



In the default color scheme Connected Controls are normally grey. However, if the connected control is not found in the audio middleware project it will be colored red.

5. Middleware Data Panel

The Audio Middleware Controls panel shows a list of all the available audio middleware controls.

i To change the Middleware you use, you have to use a CVar:

- `s_AudioImplName = CryAudioImplSDLMixer` for SDL Mixer (this is the default middleware)
- `s_AudioImplName = CryAudioImplFmod` for Fmod Studio
- `s_AudioImplName = CryAudioImplWwise` for Wwise

For more information about audio middleware implementations, see [this page](#).

i If no valid middleware is loaded or there was a typo in the CVar when switching to your preferred middleware, the UI of the ACE will be disabled and a message saying "Warning: No middleware implementation!" will be displayed in the top-right corner of the ACE.

Overview

This panel will change to always show the list of controls that can be connected to the currently selected control in the Audio System Controls panel.

Filtering

The list in this panel can be filtered using the Search bar at the top.

Connecting the Middleware Controls to the System Controls

To connect a Middleware Control to a System Control, simply select the desired System Control and drag the Middleware Control that you want to connect to it into the Properties panel under Connections.

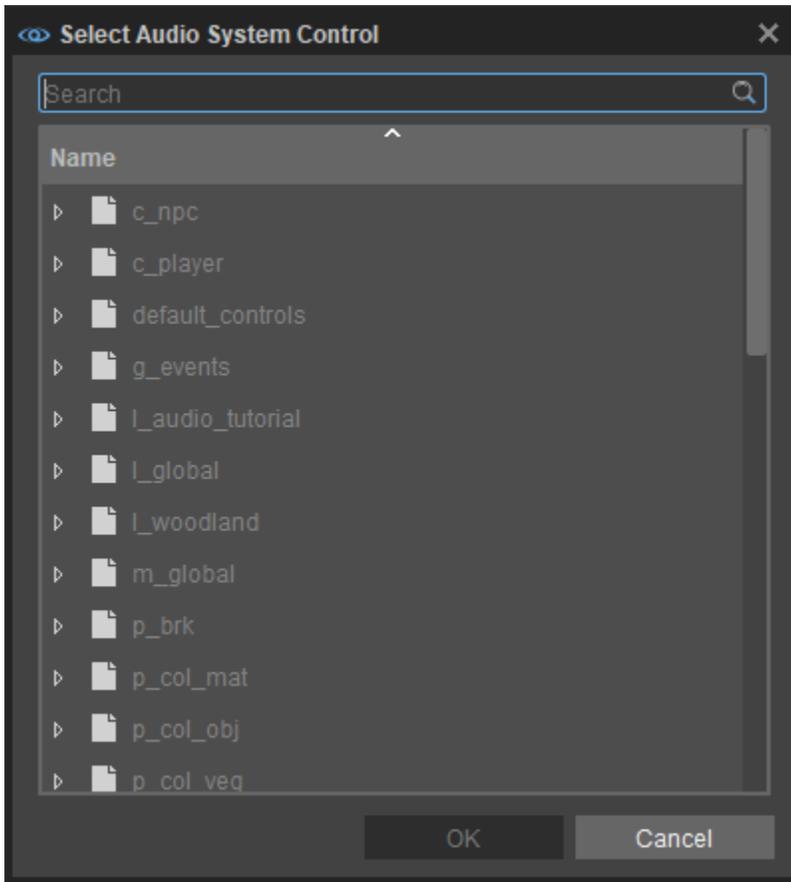
Context Menu

Depending on the Middleware you have set up, a combination of the following options will appear when right clicking a folder or Middleware Control:

Option	Description
Connections	Selects the System Control the Middleware Control is connected to, and selects its connection in the Properties panel.
Open Containing Folder	Opens the folder in the Windows Explorer that contains the selected control if the control is a physical folder or a file.
Expand Selection	Expands the selected items.
Collapse Selection	Collapses the selected items.
Expand All	Expands all items.
Collapse All	Collapses all items.

Resource Selectors

When you click a resource selector that requires an Audio System Control, like when you add a Trigger component to an entity, a window opens in which you can select the Audio System Control of your preference:



Search Bar

This works like other Search Bars, so typing anything in here will filter the Audio System Controls and only show ones that have that specific combination of letters and symbols in their name.

Context Menu

Right clicking on a folder or Audio System Control will give you the following options:

Option	Description
Execute Trigger	Executes the Trigger you right clicked on (only available when clicking on Triggers). NOTE: You can also execute a trigger by pressing space bar .
Expand Selection	Expands the selected items.
Collapse Selection	Collapses the selected items.
Expand All	Expands all items.
Collapse All	Collapses all items.

Advanced Users

Data Handling

Data for audio system controls can be found in the *.xml files in the folder `...<Project>|audio|<middleware>|ace`. Each *.xml file represents an audio library (top level item) in the Audio System Controls panel (folders are specified inside the *.xml files). Files situated in the ace root are seen and treated as files that provide data for the global data scope. However, files under `...<Project>|audio|<middleware>|ace|levels\` are seen and treated as files that provide data for the level specific data scope. In a sense these *.xml files can be understood as work units. You can spread your data across as many files as you wish or you can put all your audio-specific data into 1 single file, however that will most likely cause version control conflicts particularly if several people contribute audio work to that level. Therefore, we do advise to spread the data, but that depends on the size of the project.

Changes in Release 5.5

Migration Guide

From release 5.5 onwards, ACE files will be saved in their middleware specific folder, so each middleware implementation is completely separated by each other.

1. Create a folder called "assets" inside the current `audio/<middleware>` folder.
2. Move all soundbanks and audio files from the middleware folder to the new "assets" subfolder. For instance:
 - `audio/wwise` to `audio/wwise/assets`
 - `audio/fmod` to `audio/fmod/assets`
 - `audio/sdlmixer` to `audio/sdlmixer/assets`

Do the same for localized soundbanks if there are any. For instance:

`localization/english/audio/wwise` to `localization/english/audio/wwise/assets`

3. Backup all *.xml files of the `audio/ace` folder.
4. Remove the write protection on all xml files inside the `audio/ace` folder.
5. Remove all *.xml files from *.pak files (usually inside the `audio.pak`).
6. Launch the Sandbox and open the Audio Controls Editor from the **Tools** menu.
7. All audio libraries should now be marked as modified. Press the **Save All** button inside the Audio Controls Editor.
8. If the project contains preload requests, a dialog will appear that asks for reloading the audio system. Press **Yes**.

This completes the migration.



For cleanup remove the old **audio/ace** folder. If it contains "cryasset" files, they can be removed as well.



If ACE .xml files are write-protected they will not be automatically deleted during the migration and should be removed manually afterwards, otherwise they will get parsed again every time the Sandbox starts or the Audio Controls Editor is opened, which will increase loading time. Source controls paths and generated soundbank paths inside the middleware might need to get updated as well, depending on the project.

Audio files are loaded from the `audio/<middleware>/assets` folder and their respective localization folders `localization/<language>/audio/<middleware>/assets`.

ACE xml files are loaded from `audio/<middleware>/ace` folder.

CVars

The following CVars have been added or changed for the Audio system as of release version 5.5:

New

- **s_DebugDistance**

Limits drawing of audio object debug info to the specified distance around the active listeners. Setting this cvar to 0 disables the limiting.

Usage: `s_DebugDistance [0/...]`

Default: 0 m (infinite)

- **s_DebugFilter**

Allows for filtered display of audio debug info by a search string.

Usage: `s_DebugFilter spaceship`

Default: " " (all)

Changed

- **s_DrawAudioDebug** (changed flags)
Draws AudioTranslationLayer related debug data to the screen.

Usage: s_DrawAudioDebug [0ab...] (flags can be combined)

0: No audio debug info on the screen.

a: Draw spheres around active audio objects.

b: Show text labels for active audio objects.

c: Show trigger names for active audio objects.

d: Show current states for active audio objects.

e: Show Parameter values for active audio objects.

f: Show Environment amounts for active audio objects.

g: Show occlusion ray labels.

h: Draw occlusion rays.

i: Show object standalone files.

m: Hide audio system memory info. (new)

n: Applies a filter to inactive object debug info. (new)

v: List active Events.

w: List active Audio Objects.

x: Show FileCache Manager debug info.

u: List standalone files.

- Renamed command **s_SetRtpc** to **s_SetParameter**.

The first argument is the name of the AudioParameter to be set, the second argument is the float value to be set, the third argument is an optional AudioObject ID.

If the third argument is provided, the Parameter is set on the AudioObject with the given ID, otherwise, the AudioParameter is set on the GlobalAudioObject

Usage: s_SetParameter character_speed 0.0 601 or s_SetParameter volume_music 1.0

- **s_DrawAudioDebug g** now shows the object distance.
- Draw Occlusion Rays is now **s_DrawAudioDebug i** (was **g**).
- Draw Object Standalone Files is now **s_DrawAudioDebug j** (was **i**).
- Renamed **s_ShowActiveAudioObjectsOnly** to **s_HideInactiveAudioObjects**.
- Deleted deprecated cvars **s_AudioTriggersDebugFilter** and **s_AudioObjectsDebugFilter**.