
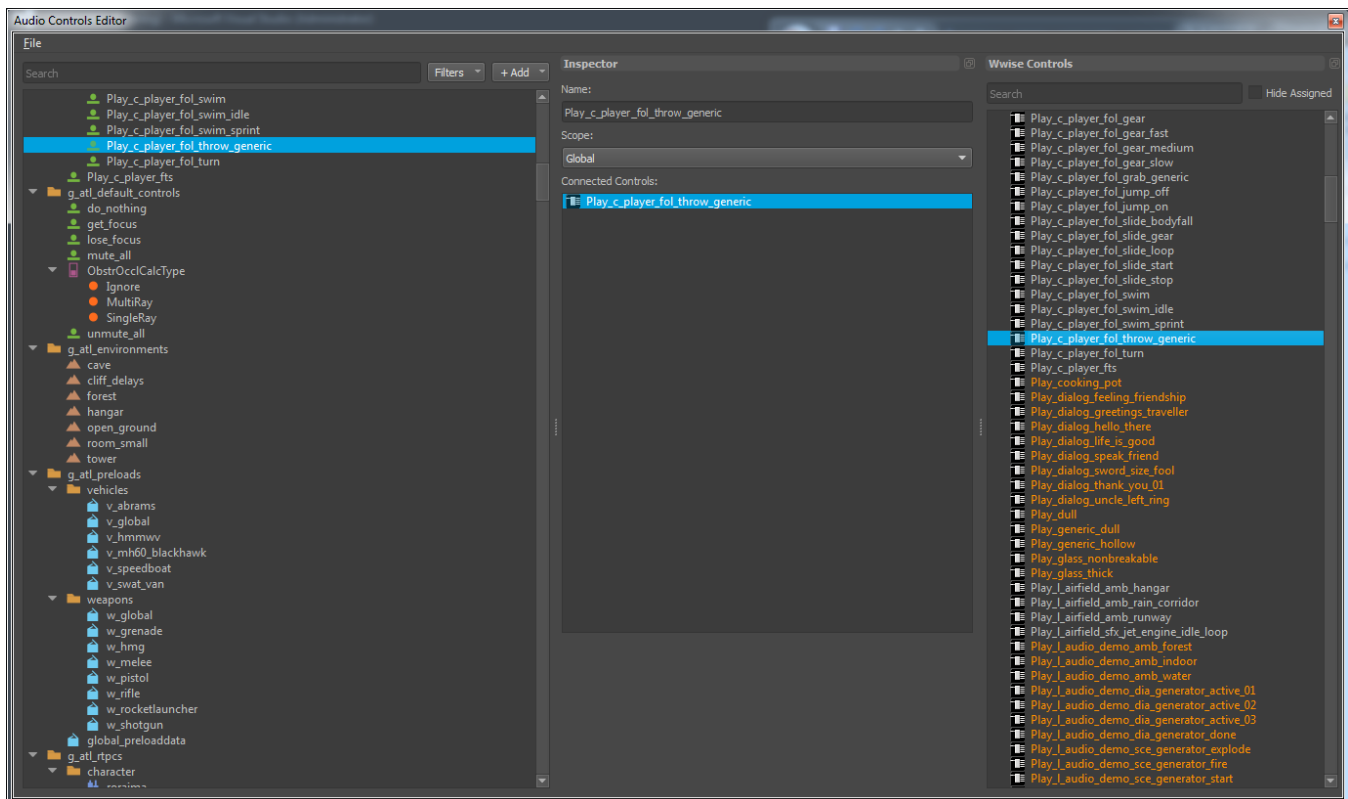


- Overview
- ATL Controls Panel
 - Overview
 - Adding Controls/Folders
 - Moving Controls
 - Edit Controls
 - Deleting Controls
 - Filtering
 - Previewing Triggers
- Inspector
 - Overview
 - Connecting Controls
- Middleware Controls Panel
 - Overview
 - Filtering
- Advanced Users
 - Data Handling
 - Version Control

Overview

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

You can find the ACE inside the Sandbox Editor by going to the menu found in View -> Open View Pane -> Audio Controls Editor or by clicking on the  button on the toolbar.



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

The ACE is composed of three main panels as shown in the following image: the **ATL Controls** panel, the **Inspector** and the **Middleware Controls** panel.

ATL Controls Panel

Overview

Shows all the controls in your CRYENGINE project. It allows you to create, move and delete the 5 different types of controls available ([Triggers](#), [RTPCs](#), [Switches](#), [Environments](#) and [Preloads](#)). You can also group your controls in folders which can be useful when organizing large projects.

Adding Controls/Folders

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

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

Moving Controls

To move controls between folders just use *drag and drop*. You can also *drag and drop* entire folders which will move their content along with it. Only folders are allowed to be moved to the root level, all Audio Controls must have a folder as a parent.

Edit 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 [Inspector](#) panel. The same functionality can be accessed by right clicking the control that is required to be modified.

NOTE:

An Audio Control name can only exist once per data scope and control type. For example, if you named an ATL-Trigger **my_trigger1** and assigned it to the Global data scope, there cannot be another ATL-Trigger anywhere in your project with the same name. You will notice that as the ACE will automatically prevent creation of another ATL-Trigger with the same name by renaming the control to **my_trigger1_1**, **my_trigger1_2** and so on. However, if triggers are assigned to Levels only it is totally fine to have the same name in each of those scopes as they are confined to that scope. Then again, you cannot have the same name inside of the Level or global data scope.

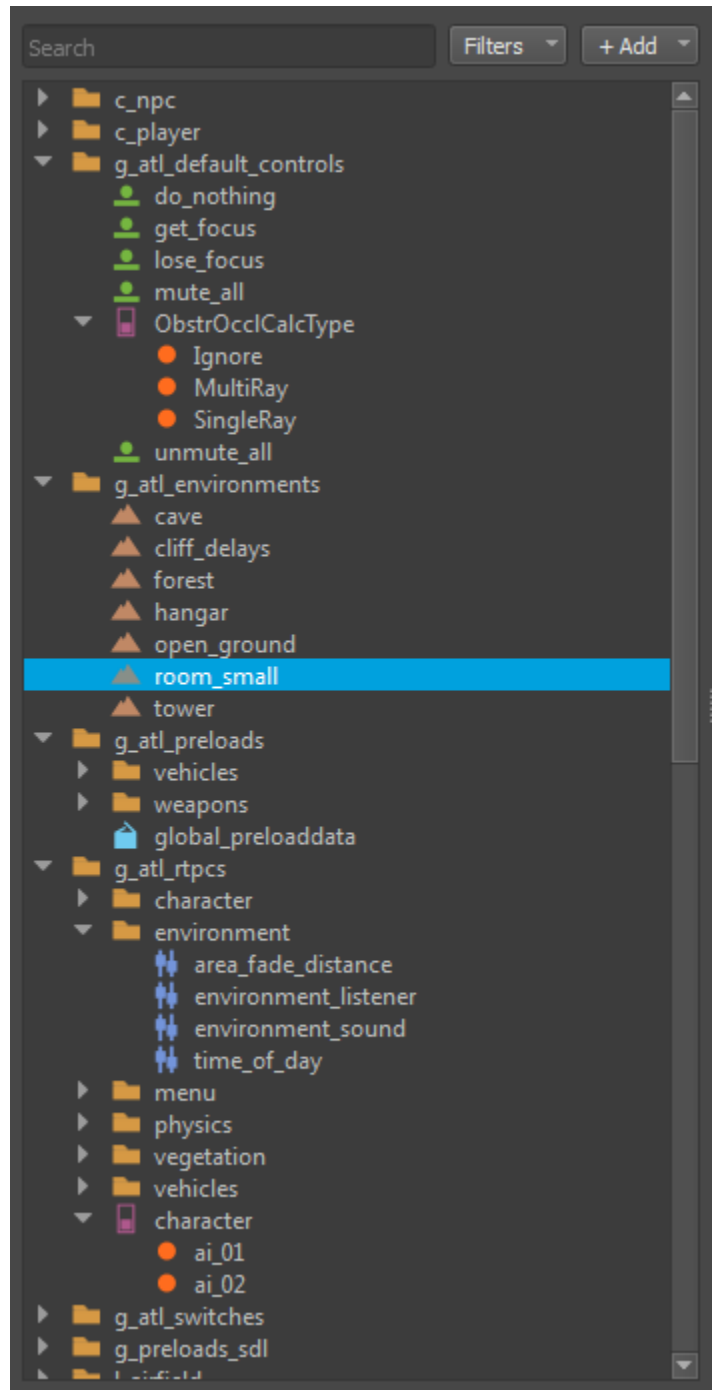
Deleting Controls

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.

Filtering

You can filter the list of controls by using the **Filter** button and toggling the visibility of the control types you want to see. Filtering the control list by type will also affect the controls shown in the [Middleware Controls](#) panel.

You can also use the search bar to filter the list by name.



ATL Default Controls

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

ATL-Trigger

- **get_focus** - is sent if the application window gets focus
- **lose_focus** - is sent if the application window loses focus
- **mute_all** - is sent by the Sandbox's "Mute Audio" button if not already muted
- **unmute_all** - is sent by the Sandbox's "Mute Audio" button if **mute_all** was sent beforehand
- **do_nothing** - does nothing, is used to override the automatic stopping of StartTrigger instances by being applied to the StopTrigger property

ATL-RTPCs

- **object_speed** - is updated on Audio Objects that have **object_velocity_tracking** enabled

ATL-Switches

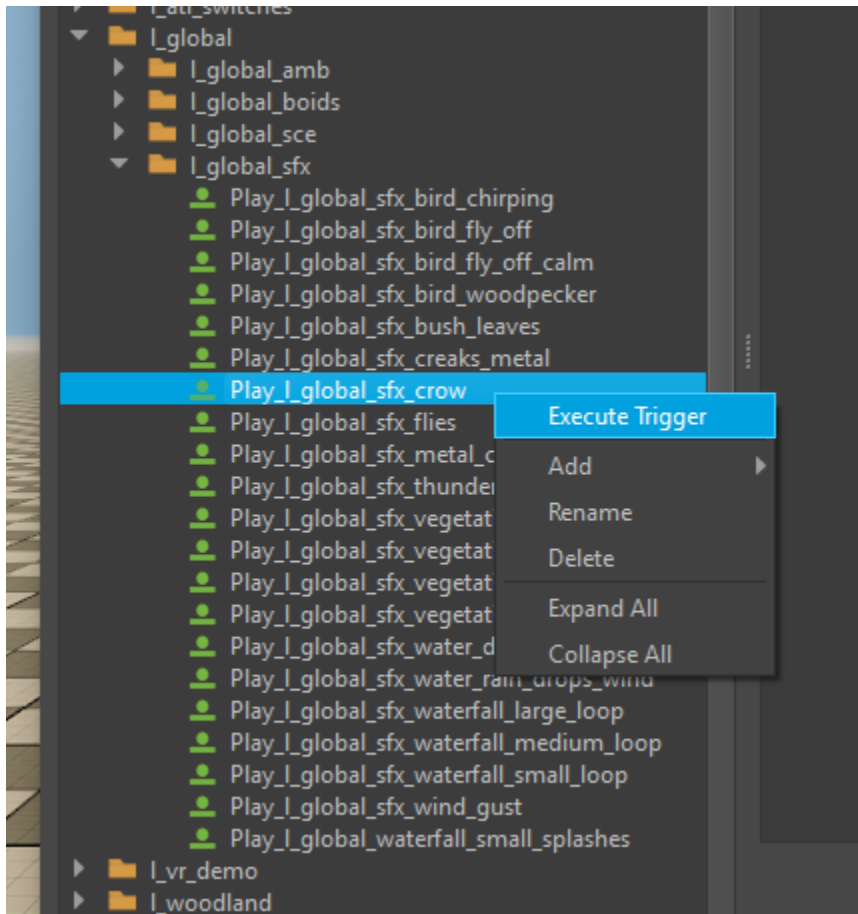
- **ObstrOcclCalcType** - is used to determine whether an Audio Object calculates obstruction and occlusion values
- **object_velocity_tracking** - is used to determine whether an Audio Object should calculate its velocity when moving and updating the **object_speed** RTPC

In the default color scheme: ATL Controls connected to middleware controls are colored grey. Those not connected to middleware controls are colored orange.

Previewing Triggers

It is possible to preview an ATL-Trigger in two ways.

1. Right click the trigger to open its context menu and choose "Execute Trigger".

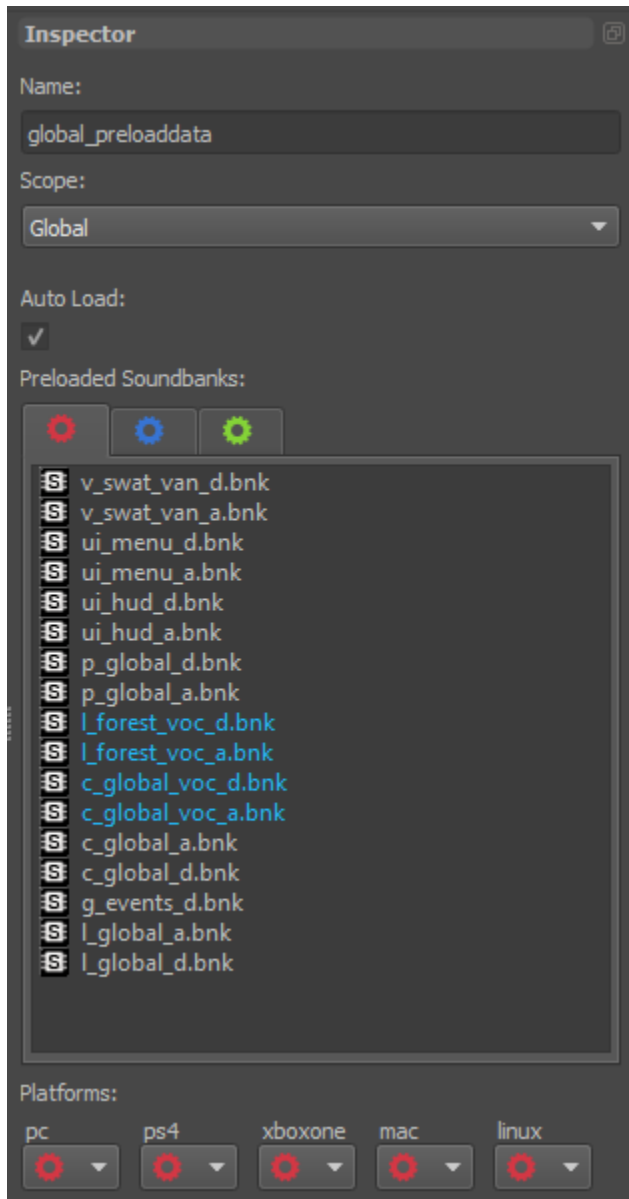


2. Hit the spacebar on the selected ATL-Trigger.

NOTES:

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

Inspector



Overview

The Inspector panel shows and allows you to edit all the properties of the control currently selected in the [ATL Controls panel](#) including making connections to any of the matching middleware controls.

The properties that can be edited in the Inspector are:

Property	Description
Name	The name of the control. This can also be edited in the ATL Controls panel .
Scope	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.
Auto Load	<i>(Only available for preloads)</i> If auto load is set, the elements preloaded with this control will be reference counted (only one copy of them is created which is shared between all its users).
Preloaded Soundbanks	<i>(Only available for preloads)</i> The Sound banks connected with a preload can be different for the different platforms your project is targeting. In this field you can add different soundbanks to different groups and then in the <i>Platforms</i> field (see below) you can choose which group to load for each platform you are targeting.
Platforms	<i>(Only available for preloads)</i> Allows you to set which group of soundbanks (defined in the <i>Preloaded Soundbanks</i> field, see above) to load for each platform. NOTE: that you can share a group between several platforms, for example you can have one group for PC and another for PS4 or Xbox One.

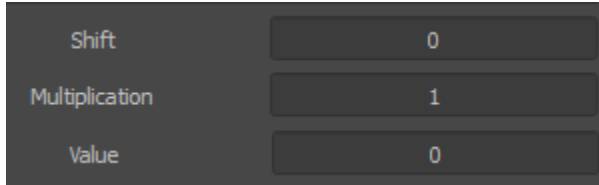
Connected Controls

This panel contains all the middleware controls connected to your ATL control. Please take a look at the [Connecting Controls](#) section for more details.

Connecting Controls

To create new connections between ATL controls and middleware controls just drag the control from the Wwise controls panel into the Connected Controls area. A middleware control can also be dragged directly into the ATL Controls panel, this will create a new control with the name of the middleware control and automatically connect both of them.

Connection Properties



RTPC Connection

Selecting a connection between RTPC's will show two additional properties in the Inspector. *Shift* offsets the value of the Wwise RTPC, *Multiplication* multiplies the value of the Wwise RTPC.

Switch-RTPCs Connection

ATL switch states can also be connected to RTPC's, so that when they are selected the corresponding Wwise RTPC value is modified. To specify this value just select the connection and edit the *Value* property that becomes available in the Inspector.

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

Middleware Controls Panel

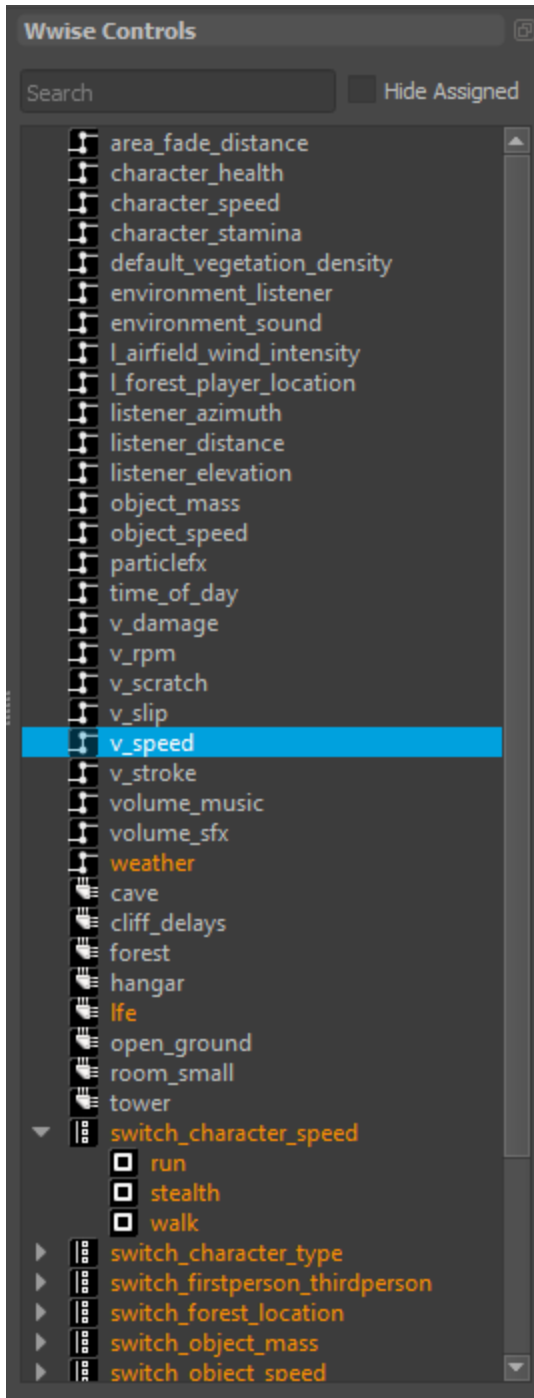
Overview

Shows a list of all the middleware controls. This panel will change to always show the list of controls that can be connected to the currently selected control in the [ATL Control](#) panel. If a folder is selected in the ATL Control panel all (non-filtered out) controls will be displayed.

Filtering

The list in this panel can be filtered using the Search bar at the top. The list of controls is also restricted to the control types enabled with the *Filter* button in the ATL Control panel.

Checking the *Hide Assigned Controls* will hide all controls which have already been connected to an ATL control.



In the default color scheme Connected Middleware Controls are colored grey. Non connected Controls are colored orange.

Advanced Users

Data Handling

Data for ATL controls can be found in the .xml files in the folder "...<Project>\Libs\GameAudio". Each .xml file represents a top level folder in the ATL Controls panel (subfolders are specified inside the .xml files). Files situated in the "GameAudio" root are seen and treated as files that provide data for the global data scope. However, files under "...<Project>\Libs\GameAudio\Levels\" are seen and treated as files that provide data for the Level specific data scope. In a sense these xmls can be understood as workunits. 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.

Version Control

The ACE can be used along side our Sandbox version control plugins. If properly setup any changes to the controls will result in the plugin checking out and marking for add/delete the underlying files.