
Lesson 2: Designing a Soundscape

Adding Sounds to a Project	56
Using a Single Sound For Multiple Applications	62
Changing Object Properties	65
Using the Source Editor	68
Related Video	70
Applying Multiple Actions to a Single Event	70
Using Randomization	78
Importing an Audio File Folder	78
Using Controlled Randomization	83
Randomizing Properties	86
Related Video	88
Granulate Your Sounds	88
Creating a Sequence	97
Related Video	103
Using Silence	103
Verifying Your Work	109

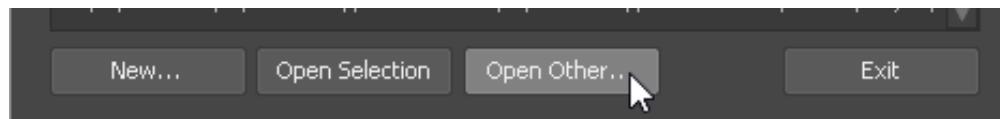
Lesson 2: Designing a Soundscape

Now that you've attached a blast sound to the animation of your character firing a shotgun, you'll now look at other parts of the visual sequence that accompanies the sound and think about what else you can do to enhance that experience. As you work through the steps that need to be done, you'll also be looking at how to do as much as possible with as little as possible, for this is what defines a game audio designer's worth. This is because while modern game console systems have incredible power, that power is shared with systems that support graphics, physics, and artificial intelligence. Also consider that many games are played on mobile devices where system resources aren't as vast. Depending on the game's technical restraints, audio may be given a small budget with regards to memory and processor usage. Because of this, an audio game developer makes their mark by learning how to fuse great sound and creativity within the technical limitations provided. In many cases, it's the designer's creativity that provides unique and interesting solutions for working within technical constraints. Fortunately, Wwise provides a vast set of tools specifically designed to help you rise to this kind of challenge.

Adding Sounds to a Project

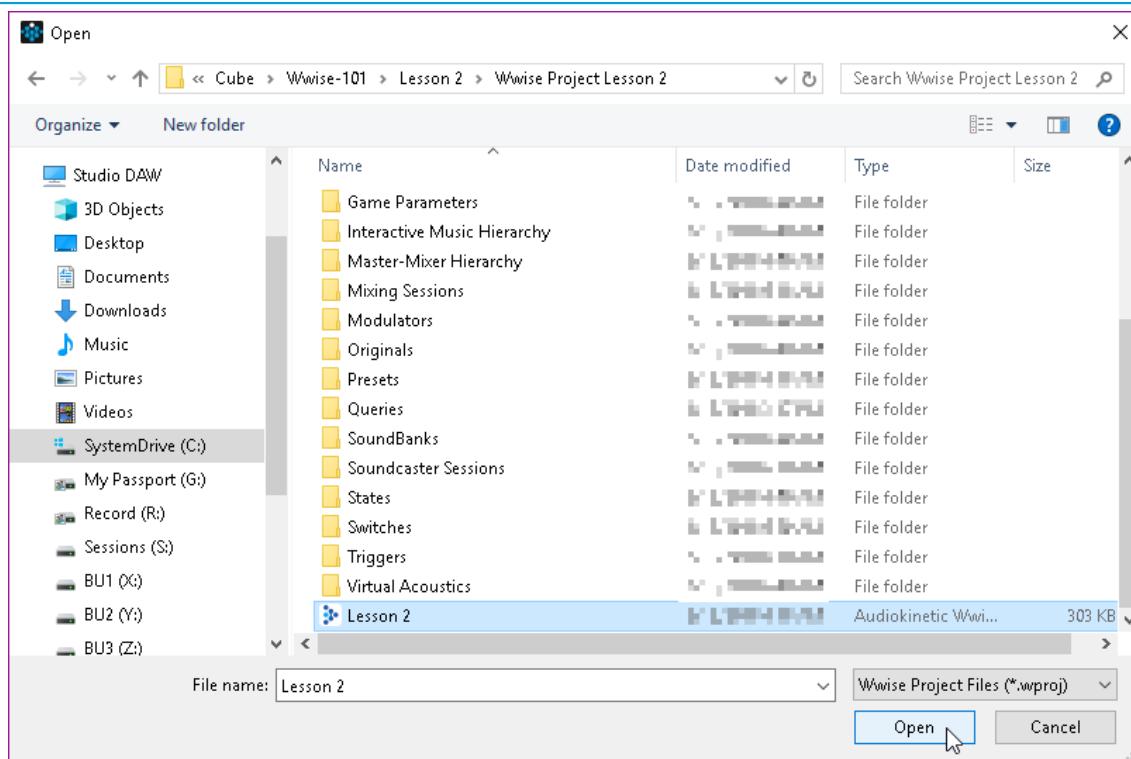
For this lesson you'll start by using a pre-saved project file that already has the shotgun blast sound Effect you used in the last lesson.

1. Launch Wwise and click **Open Other** at the bottom of the window of the Project Launcher.



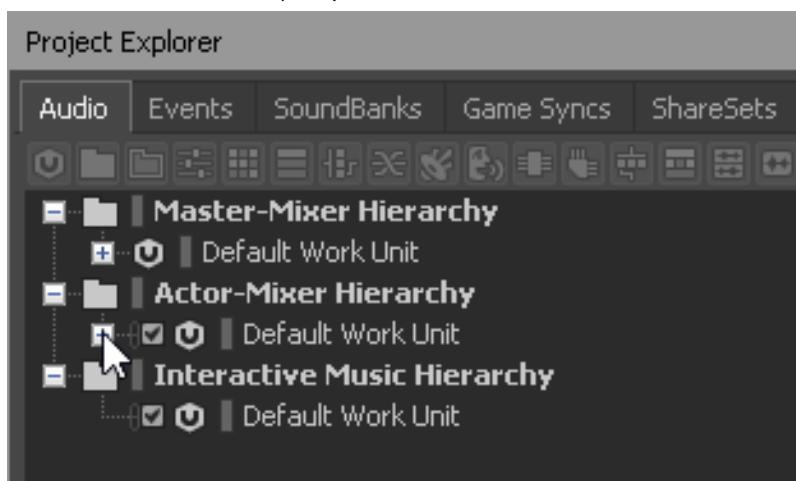
2. Navigate to your **WwiseProjects > Lesson 2** folder.

Lesson 2: Designing a Soundscape



A Wwise project is not a singular file, but instead a specific structure of many different folders and the files contained within. In most cases, you won't need to directly work with the content inside these folders as the necessary items will automatically be created and updated as you build a project. Projects are launched from a Wwise Project file which is located on the same level as these folders.

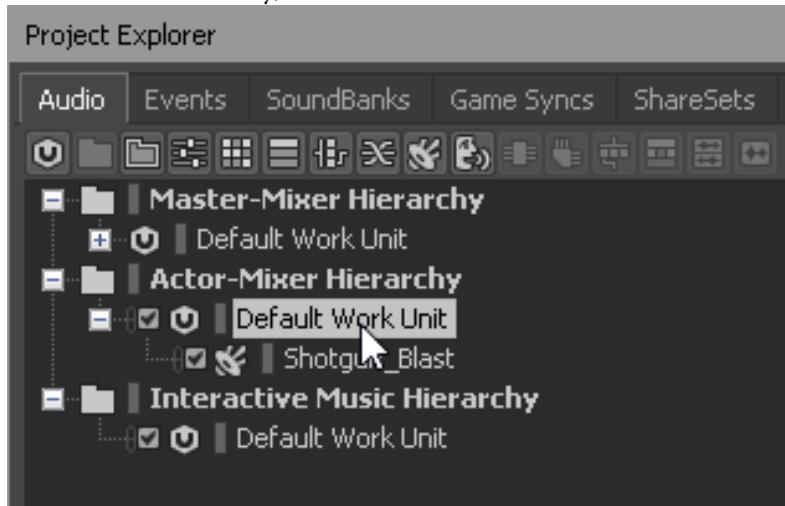
3. Select the Lesson 2 project file found within the Lesson 2 folder and click Open.
4. In the Actor-Mixer Hierarchy, expand the Default Work Unit to reveal the Shotgun_Blast Sound SFX object you created in the last lesson.



Lesson 2: Designing a Soundscape

Instead of first creating the Sound SFX object and then adding a sound to it, you'll take a different approach by simply importing the audio files to the desired Work Unit, and let Wwise automatically create the necessary Sound SFX objects.

5. In the Actor-Mixer Hierarchy, select the Default Work Unit.

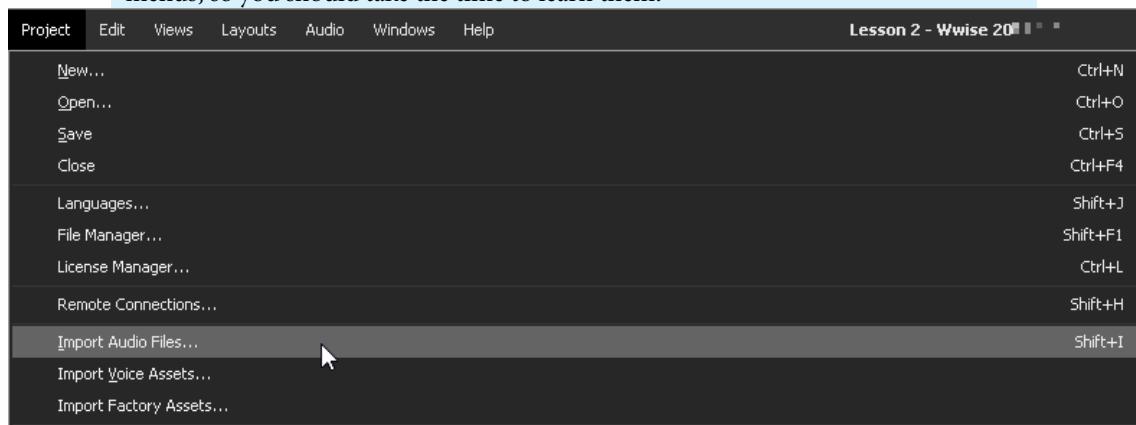


You may see that the Shotgun Blast Sound SFX object appears in blue instead of white. Blue indicates that the audio file being referenced is the original imported file, while white designates that the file has been converted. Unless the object appears in red—indicating a file is missing, the color difference does not change the operation of the exercises used throughout this tutorial. If you prefer, you can convert all of the audio files in your project at any time by going to the main menu bar and choosing Project > Convert All Audio Files. You'll learn more about Conversion Settings in Lesson 7.

6. In the main menu, choose Project > Import Audio Files or press Shift+I.

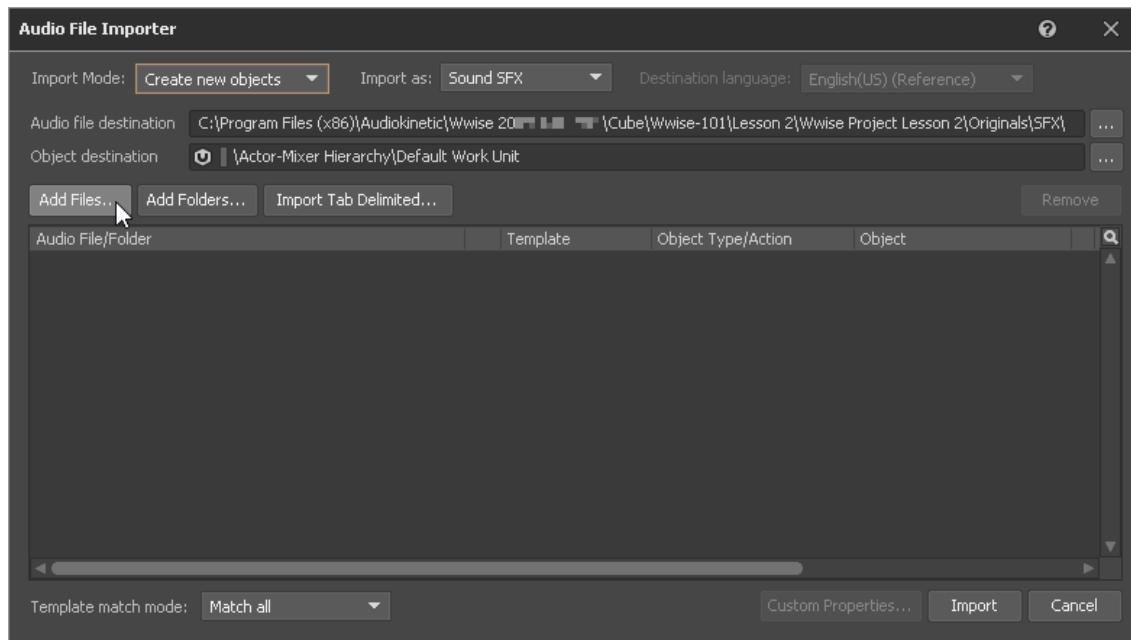


Using key commands for things you do on a regular basis will make you much more efficient as you work. Wwise displays key commands for features found in the menus, so you should take the time to learn them.



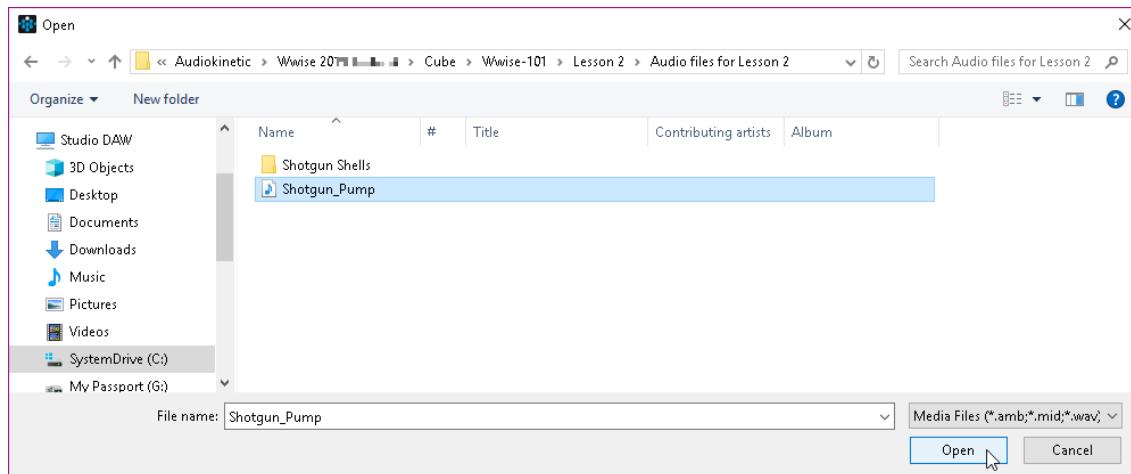
Lesson 2: Designing a Soundscape

7. In the Audio File Importer, click Add Files.



An Explorer window opens where you can navigate to the file that you want to import.

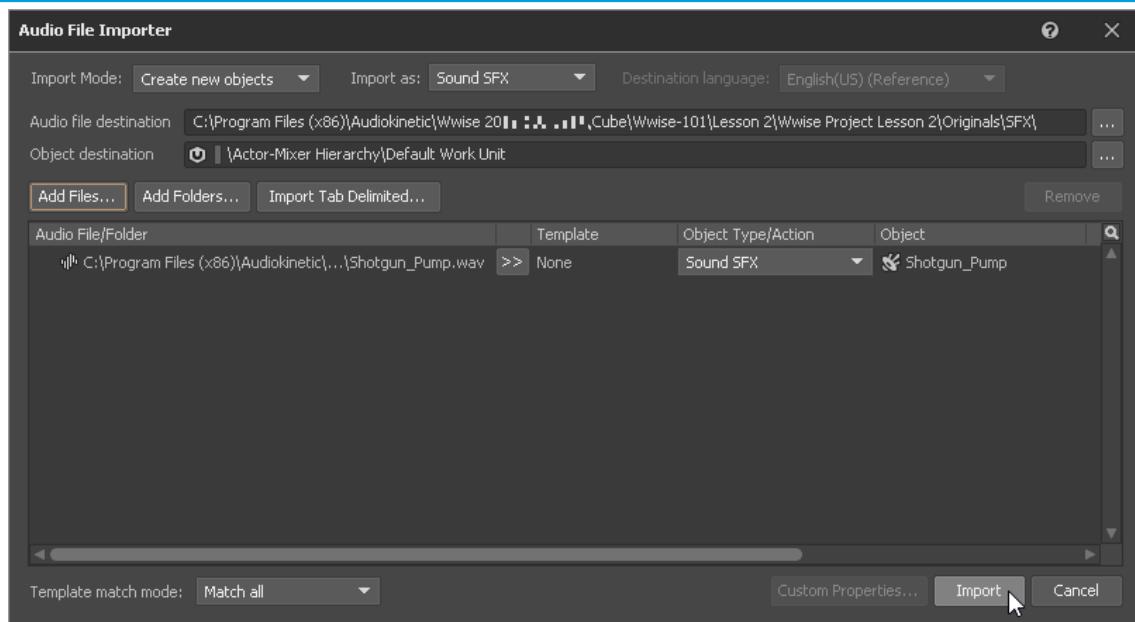
8. Navigate to your Lesson 2 folder in Wwise-101. Open the Audio files for Lesson 2 folder, choose Shotgun_Pump and click Open.



The Audio File Importer window displays the current location. You'll also see an Object Type/Action pull down menu indicating that the imported file will automatically have a Sound SFX object created to contain it.

9. Click Import.

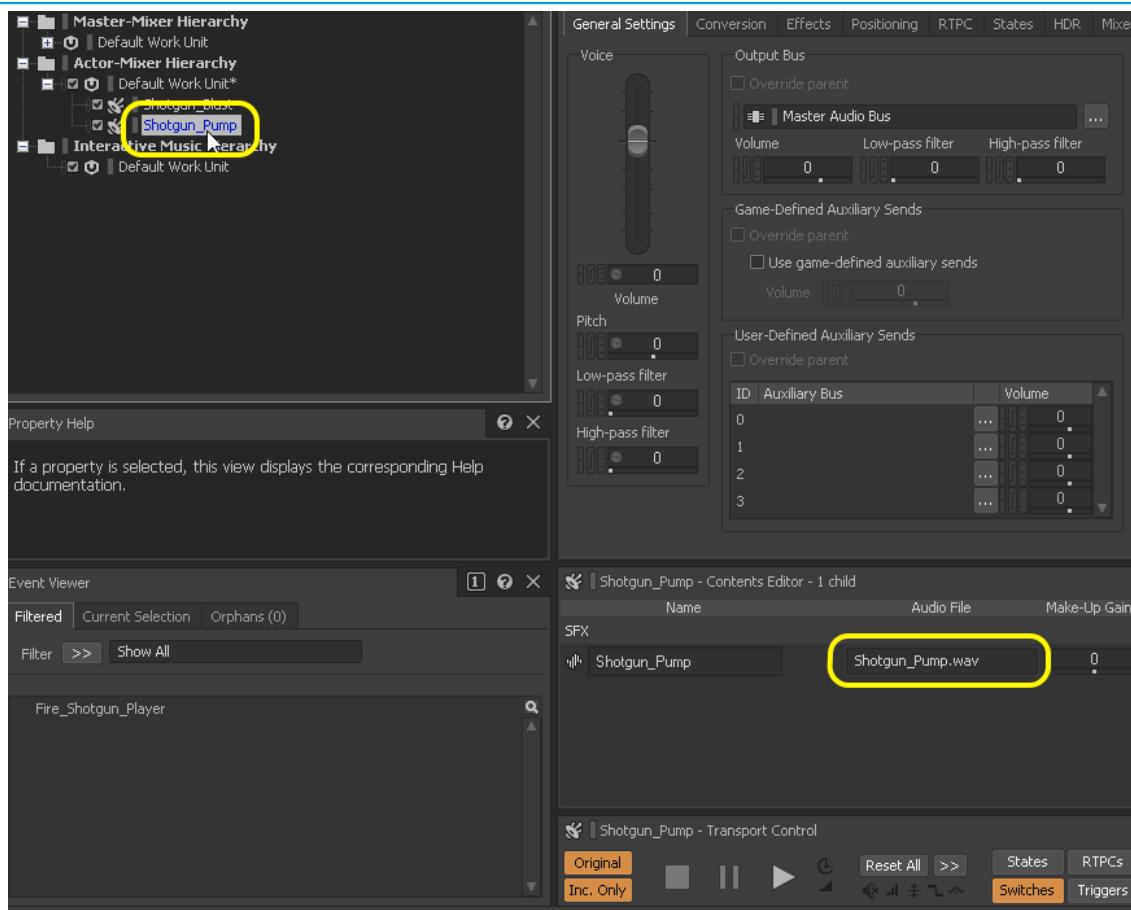
Lesson 2: Designing a Soundscape



The Shotgun_Pump Sound SFX object is displayed within the Actor-Mixer Hierarchy's Default Work Unit.

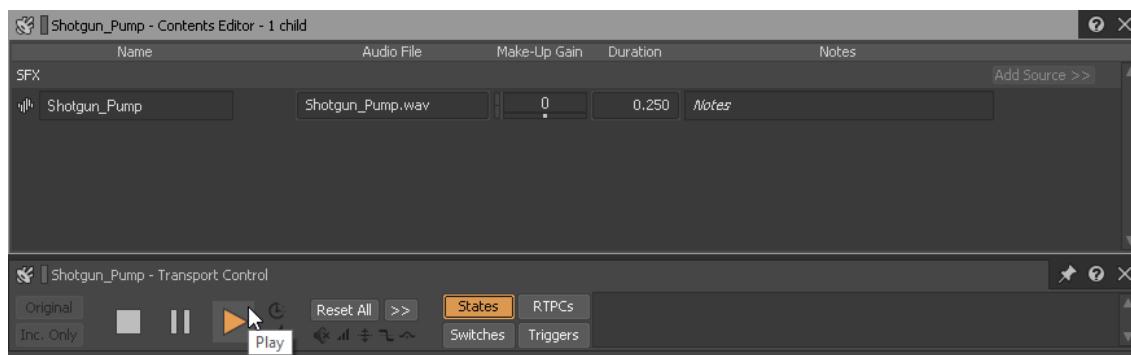
10. Click the Shotgun_Pump SFX object to view the Sound Property Editor with the parameters for the newly created object.

Lesson 2: Designing a Soundscape



In the Contents Editor (under the Sound Property Editor), the Shotgun_Pump.wav file is shown as the source that feeds the Shotgun_Pump Sound SFX object.

11. To listen to your newly imported audio file, make sure that the Shotgun_Pump Sound SFX object is selected in the Project Explorer and then click the Play button in the Transport Control View, or press the spacebar.



You hear a single mechanical click of a shotgun being pumped.

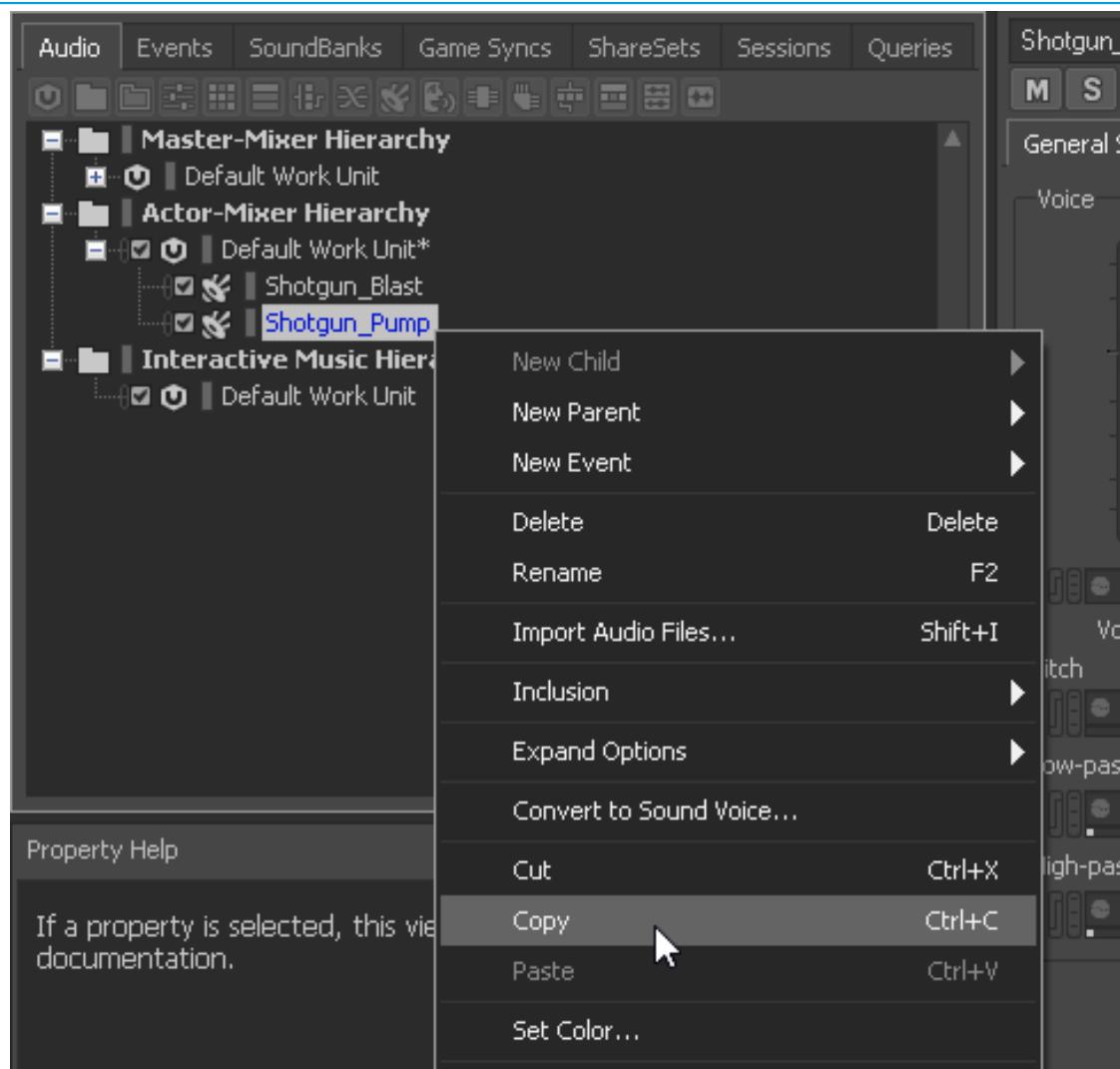
Using a Single Sound For Multiple Applications

You may have noticed that when you watch the animation sequence of the player shooting the shotgun that there are actually two phases to pumping the shotgun. First, the character pumps the shotgun inward, which on a real shotgun ejects the empty shell and then pushes the pump forward which loads the next round into the chamber. Each of these movements makes a mechanical clicking noise; however, the audio file that you just imported does not contain two clicks.

You could load a single recording of two pumps that is pre-timed to match the animation sequence, but that makes the audio file longer and therefore takes more memory when loaded into the game. Remember that preserving memory resources is something that you should always seek to achieve, so this might not be a good strategy. It's also common that the timing of animation sequences may change as the game's development progresses, so you'll take a different approach and use this single mechanical click for both the pump in and pump out sounds, using less memory and providing more timing flexibility. While you'll be using the same sound file, you're going to use two separate Sound SFX objects, one for pump in and one for pump out.

1. Right-click the Shotgun_Pump Sound SFX object and choose **Copy**.

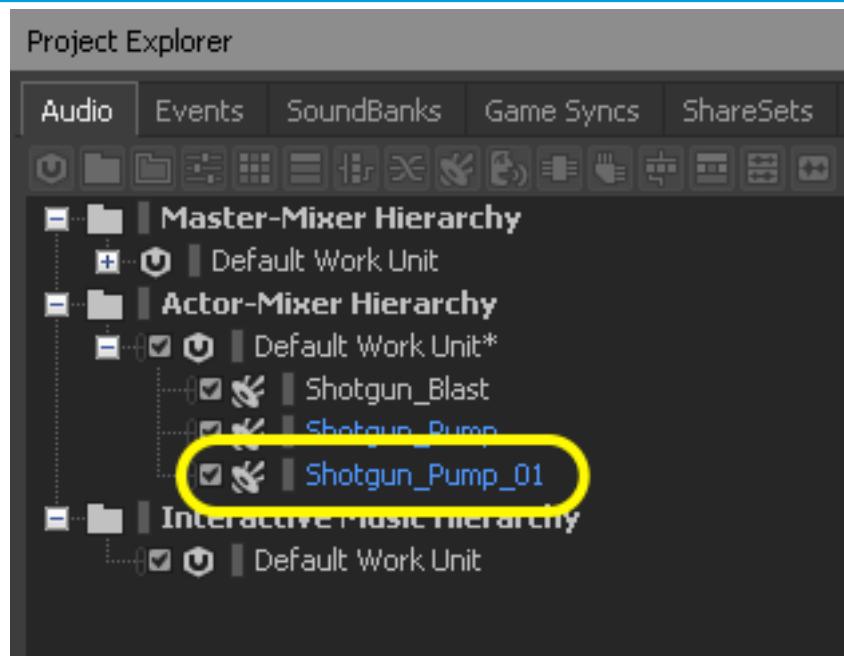
Lesson 2: Designing a Soundscape



You now need to create a copy of the object within the same Work Unit.

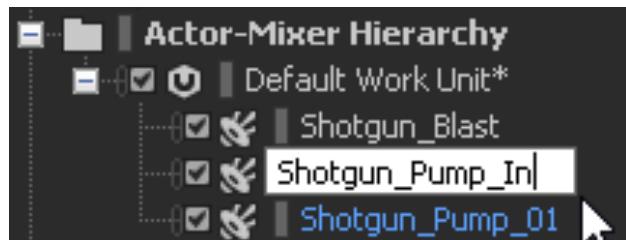
2. Within the Actor-Mixer Hierarchy, right-click the Default Work Unit and choose Paste.

Lesson 2: Designing a Soundscape



Wwise creates a duplicate of the Sound SFX object but adds an _01 extension.

One of these Shotgun_Pump objects will serve to create the pump in sound, while the other will be the pump out so it's best to name them as such.



3. Click the Shotgun_Pump object, pause, and click again to rename it to Shotgun_Pump_In, and then press Return.

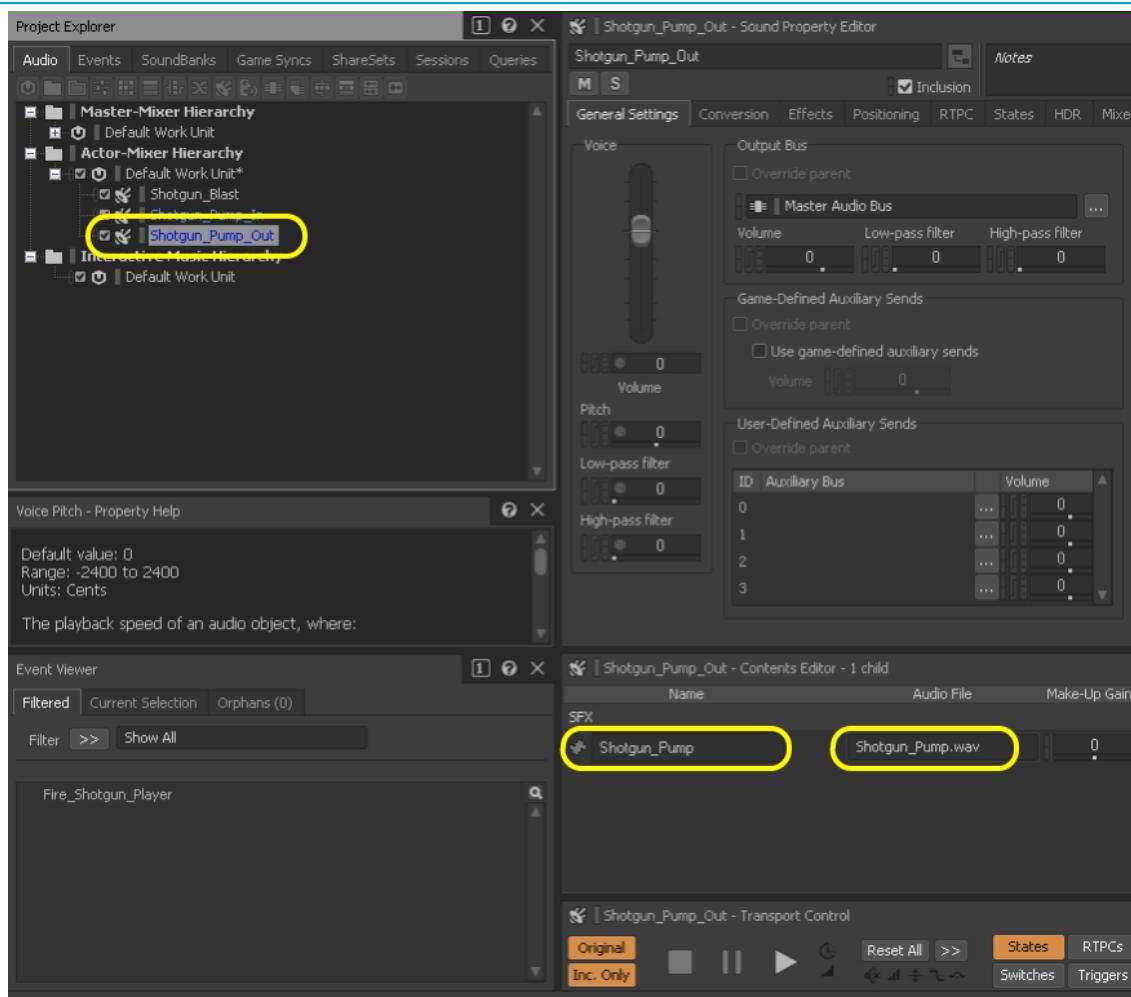


You can also rename an object by pressing the F2 key, right-clicking and choosing Rename or by directly entering into the Name field within the Property Editor.

4. Rename the Shotgun_Pump_01 object to Shotgun_Pump_Out.

The Shotgun_Pump_Out object is now selected, so you can see that this object that you copied is referencing the same *Shotgun_Pump.wav* file.

Lesson 2: Designing a Soundscape

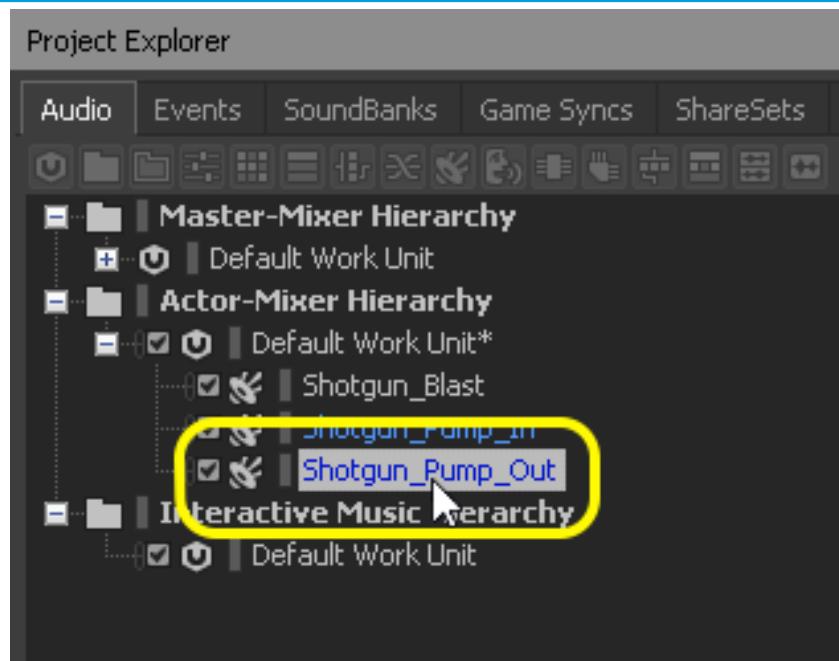


Changing Object Properties

While both of the *Shotgun_Pump* objects indicate they are referencing a file with the same name, this does not mean that the *Shotgun_Pump.wav* audio file was duplicated. It means that the same audio file is being referenced by two different sound objects, which does not double the amount of memory necessary to play these objects. It's similar to the idea of a Digital Audio Workstation where a single audio file may be placed on multiple tracks to be edited and processed in different ways. In the same way, you'll now make modifications to the properties of *Shotgun_Pump_Out* object to differentiate it from the pump in.

1. Verify that the *Shotgun_Pump_Out* object is selected and if not, select it.

Lesson 2: Designing a Soundscape



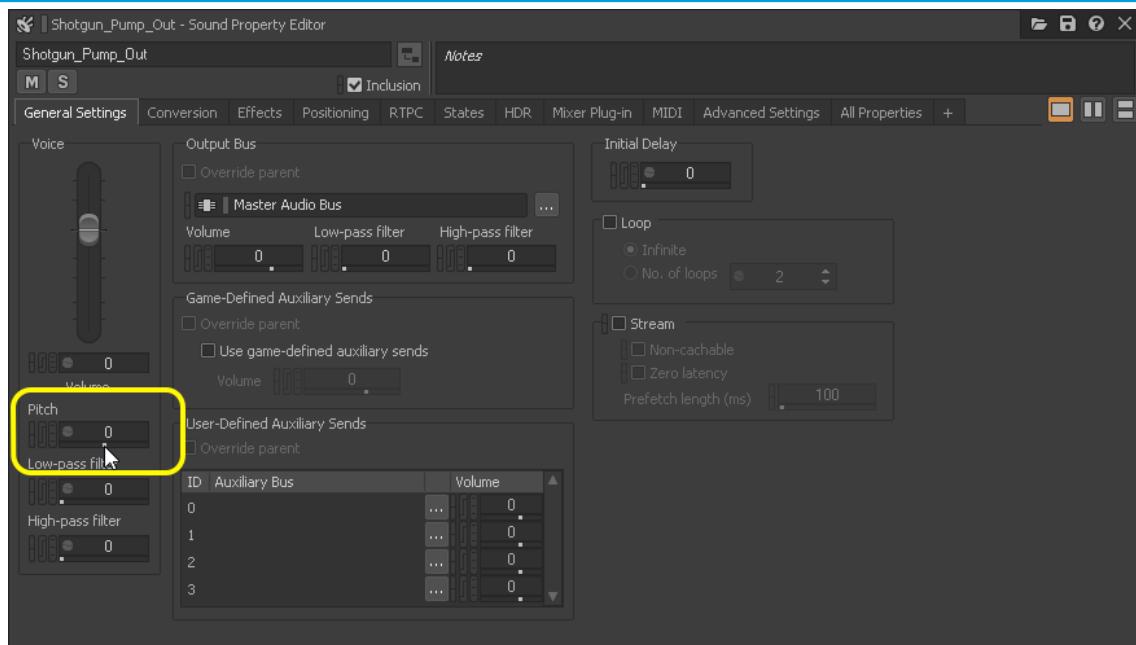
The Sound SFX Property Editor view to the right displays settings that affect how the audio file contained within the selected object will play. There are numerous parameters available that you will explore later in this lesson as well as throughout this tutorial, but one very important parameter in game audio is pitch. You'll find the pitch parameter displayed just below the volume fader. This is very convenient for the game audio designer because usually in audio production tools, the pitch control is not prominently displayed even though you use it regularly. This is especially the case in game audio where pitch adjustment is constantly used, oftentimes to create the illusion of having an abundance of different sounds using only a single sound played at varying pitches. This strategy preserves precious memory. You'll use this approach to differentiate the pump in and pump out sounds.

The pitch parameter value is expressed in musical cents. On a piano, there is 100 cents of pitch change between any two keys that are side by side, whether they're black or white keys. There's a total of 1200 cents in a musical octave. The pitch parameter default value is 0, meaning that the audio associated with the object will play back at its original pitch.

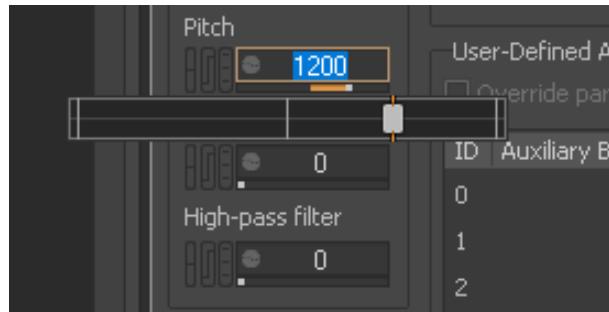
To adjust the pitch value, there's a small white dot below the pitch property that when clicked brings up a slider.

2. Click and hold the slider head (white point) just below the current pitch value.

Lesson 2: Designing a Soundscape



You can now adjust the slider to the left or right to adjust the playback pitch.



On the left side of the layout, you can see the Property Help view. This contextual view will conveniently display relevant information for any property you select. For the pitch property it confirms that cents are the unit type and that you can adjust the value within a range of -2400 to 2400, a total span of four octaves.



The language displayed in the Property Help view can be changed. In the main menu choose **Help > Documentation** and select the desired language.

Start by adjusting the pitch value to around 1200, or one octave, but it's ok if you're not able to settle on that specific value.

3. Set the Pitch property to approximately 1200 and press spacebar to hear the change.

Lesson 2: Designing a Soundscape

The Shotgun_Pump audio file plays an octave higher, resulting in a sound that plays twice as high and twice as fast. This is obviously too much, so now set the value to something that's a little more believable, like 200. If you already know the specific value you want you can directly enter it in.

4. Click the Pitch properties current value, type 200 and press Return.



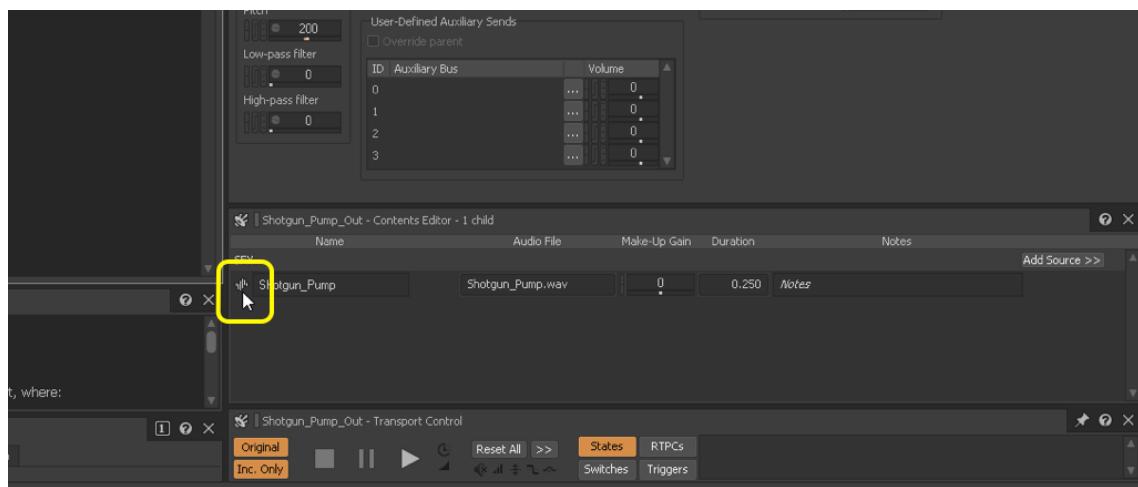
5. Play the object once more to hear the change.

Now you have a pump out that's a bit more believable and slightly different from the pump in.

Using the Source Editor

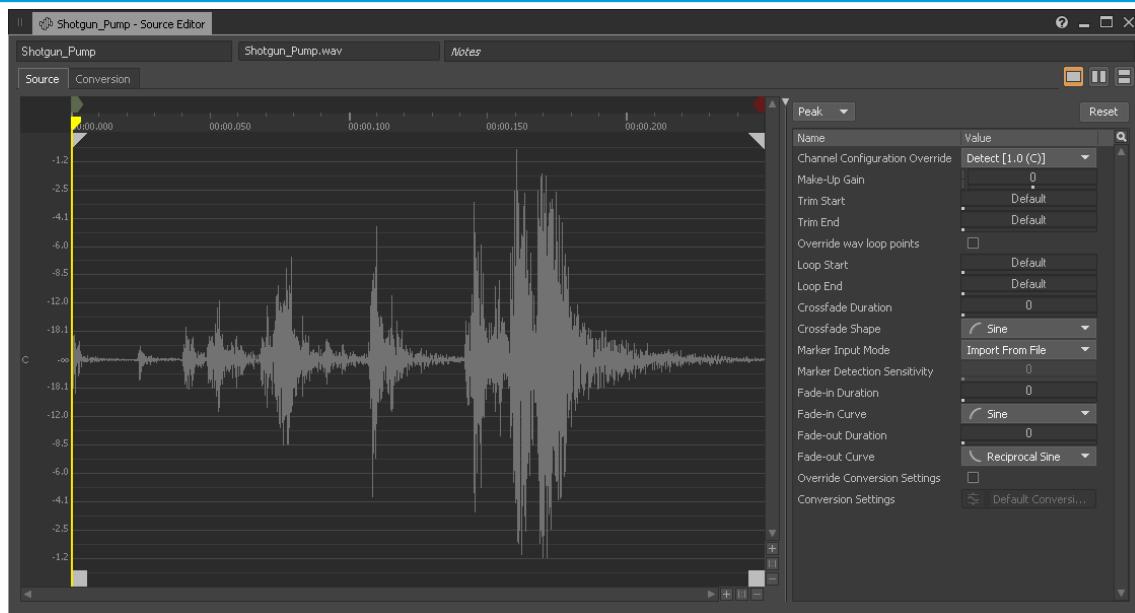
Although the pump out sound is playing at a slightly higher pitch, it's still identical in every other way. While it's a short sound, there's still a rhythmic pulse created by the pump mechanism's mechanical action. People tend to notice these subtle cues, so you can give the pump out its own distinct feel by slightly altering where the sound starts playing from. This is accomplished using the Source Editor.

1. Make sure the Shotgun_Pump_Out SFX object is still selected and then in the Contents Editor, double-click the audio source icon for the Shotgun_Pump audio file.



The Source Editor opens.

Lesson 2: Designing a Soundscape

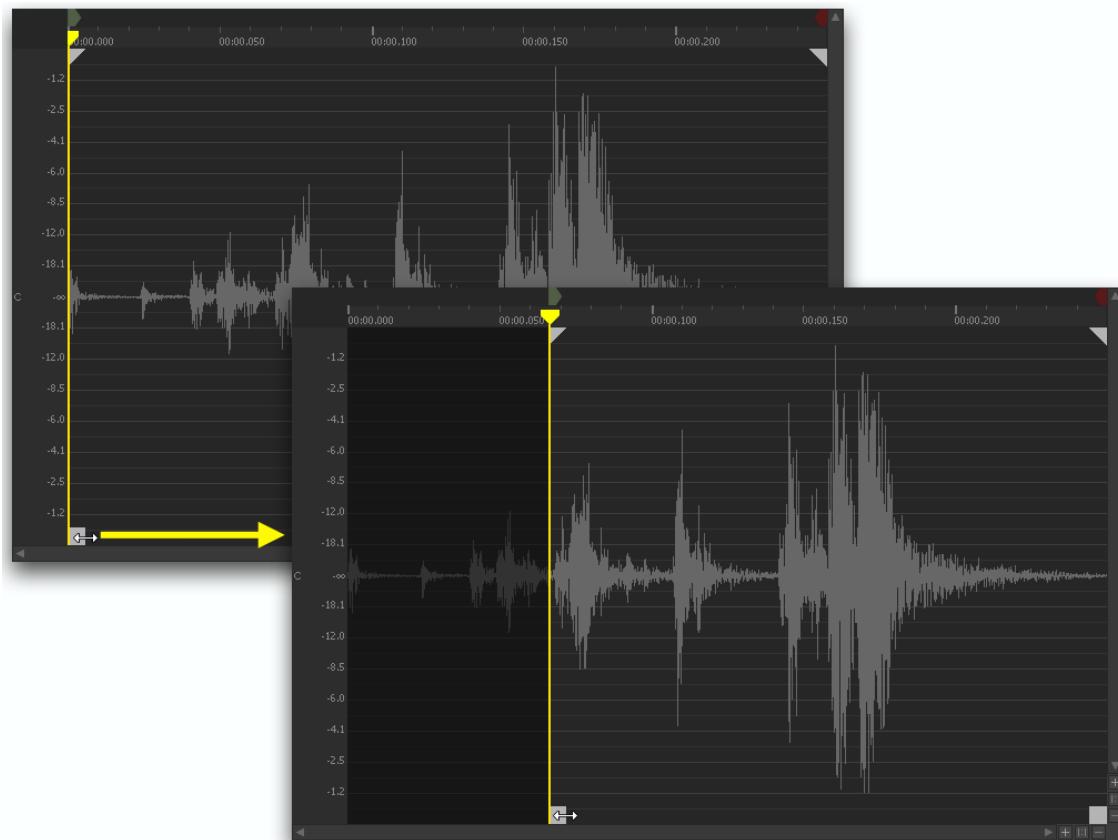


The Source Editor provides a way to look at details related to the source audio generator. Wwise provides several methods of generating sound including tone generators and the specialized SoundSeed synthesizers. Clicking the Source Editor when these tools are being used displays the interface for these particular generators. In this case, since the source audio generator is an audio file, a waveform view of the audio signal is shown. This display also serves as a simple waveform editor, which lets you adjust where playback starts and ends.

For the pump out, you're going to start playback at a later time in the audio file. You can easily do this by dragging the Trim Start control to the position that you want playback to start from. By reading the waveform, you can see where the rhythmic pulses of the mechanical noise are. Ideally, you want to begin playback just before one of these sections.

2. Drag the Trim Start control about a fourth of the way into the waveform, just before one of the peaks in the waveform.

Lesson 2: Designing a Soundscape



This change to the start position only occurs when the audio file is played from the Shotgun_Pump_Out sound object and doesn't change the way you heard the Shotgun_Pump_In object. In summary, the Sound SFX object may use the same source audio file, but each can have a unique start time, providing differentiation between the objects.

3. Play the Shotgun_Pump_Out object to hear the difference, then close the Source Editor window.

Related Video

[Wwise-101-02 - Source Editor](#)

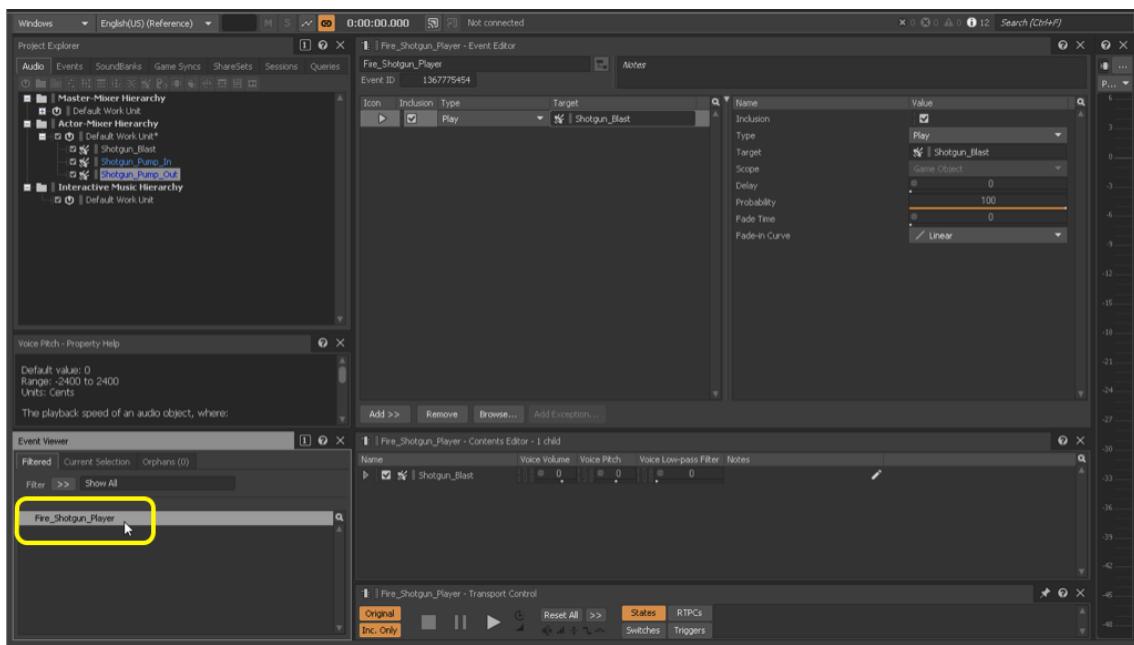
Applying Multiple Actions to a Single Event

Now that you've set up the pump sounds, you need to get them to play at the right time. Visually, when the player fires the shotgun, you see the blast, followed by a pump-in and finally the pump-out action. You could ask the game engine programmer to provide separate pump-in and pump-out game calls so you could then create corresponding Events as you did with the original Fire_Shotgun_Player Event used for the shotgun blast. Or, you could look to see if there's a way to achieve what you need without programmer interaction, because this saves precious time for both you and the programmer. Since the shotgun blast and the pumps are all part of the same visual animation and that animation never

Lesson 2: Designing a Soundscape

changes, the timing between the blast and the pump-in and pump-out visuals are predictable. You already have the Fire_Shotgun_Player Event for the shotgun blast, so you simply need to trigger the pump sounds to happen at a pre-determined time after that Event. This can be accomplished by delaying the Play actions.

1. In the Event Viewer, select the Fire_Shotgun_Player Event.



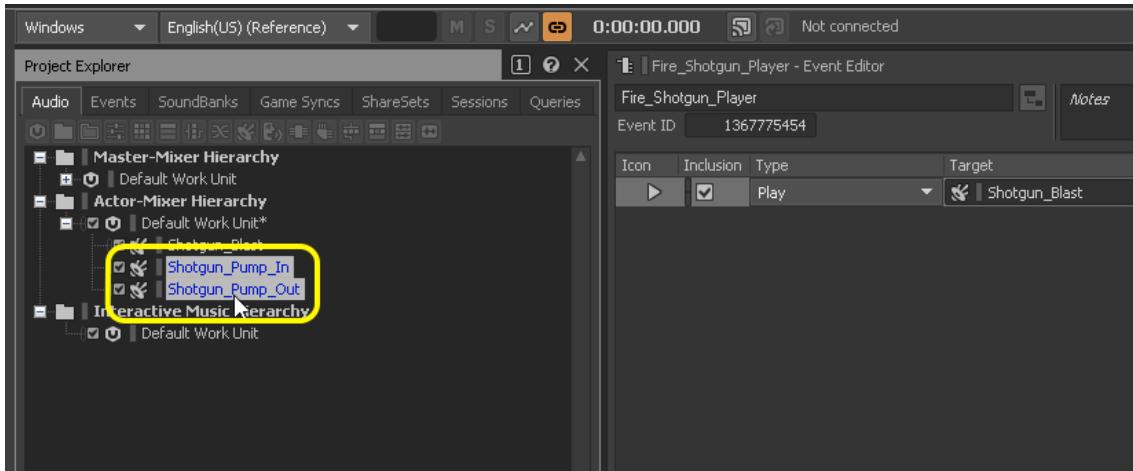
You can see in the Event Editor the Play Shotgun_Blast action you configured in the last lesson. There's no practical limit to how many actions can be carried out for a single Event, so you'll simply add the Shotgun_Pump objects into the Event Editor's Actions pane. This time, however, instead of first creating a Play action and then adding the object to be played, you'll simply drag the objects from the Project Explorer into the Actions pane. To do this you need to select both of the Shotgun_Pump objects.

2. Press and hold the Shift key and select the Shotgun_Pump_In and Shotgun_Pump_Out objects.

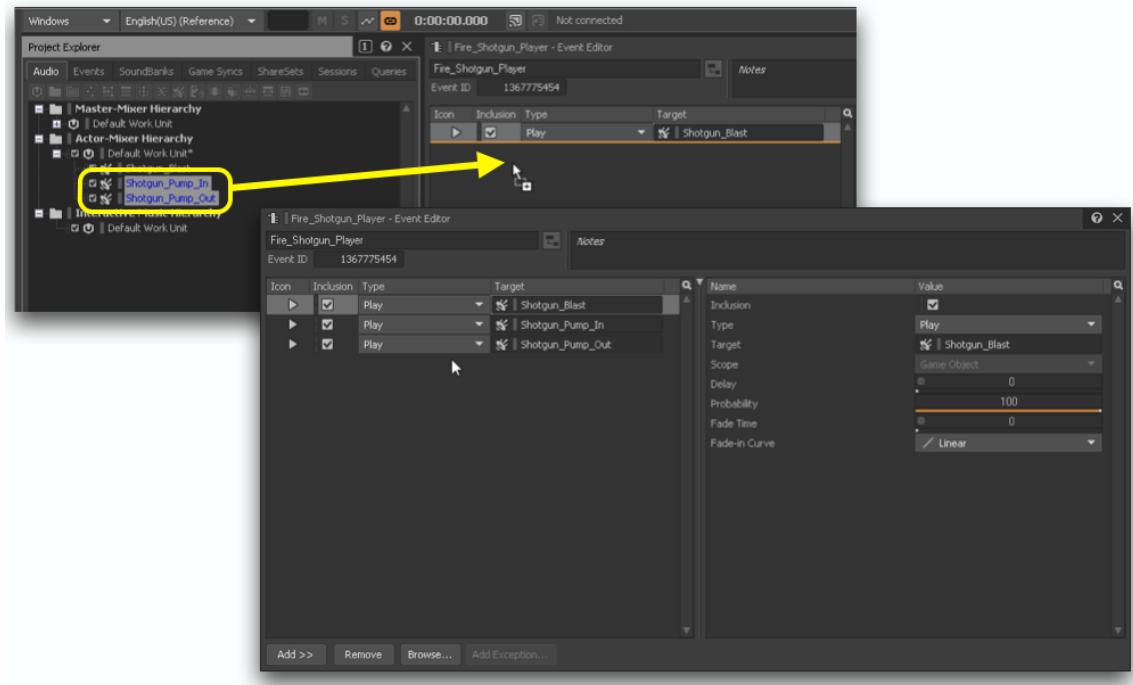


If you don't hold down shift first, the moment you click on one of the Shotgun_Pump objects, the Event Editor will change to the Property Editor and you won't be able to complete the next steps.

Lesson 2: Designing a Soundscape



3. Drag either of the selected objects to the Actions pane in the Event Editor.



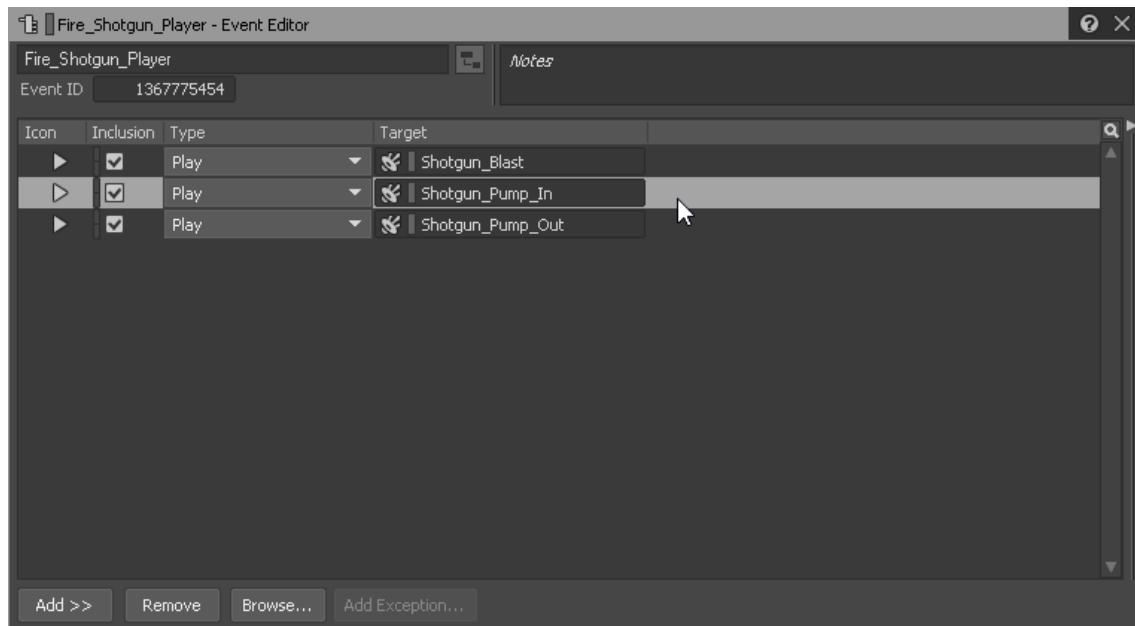
The objects are now added to the Event Editor below the Shotgun_Blast. While these actions are displayed as a list, it's important to know that these actions are not carried out sequentially, one after the other, but instead happen simultaneously unless you make some additional changes.

4. Play the Fire_Shotgun_Player event object.

Notice that the blast and the pump sounds play at the same time. To get the pump sounds to match the animation sequence, you'll need to delay each of them with specific values. Start by delaying the pump in sound.

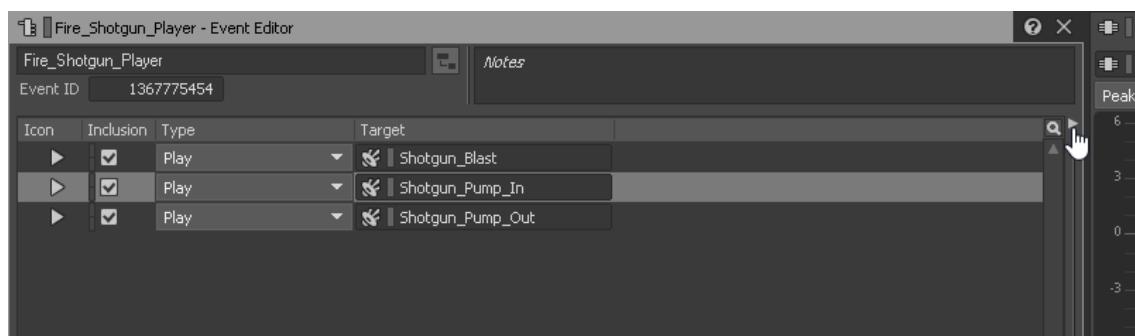
Lesson 2: Designing a Soundscape

5. In the Event Editor choose the Shotgun_Pump_In.



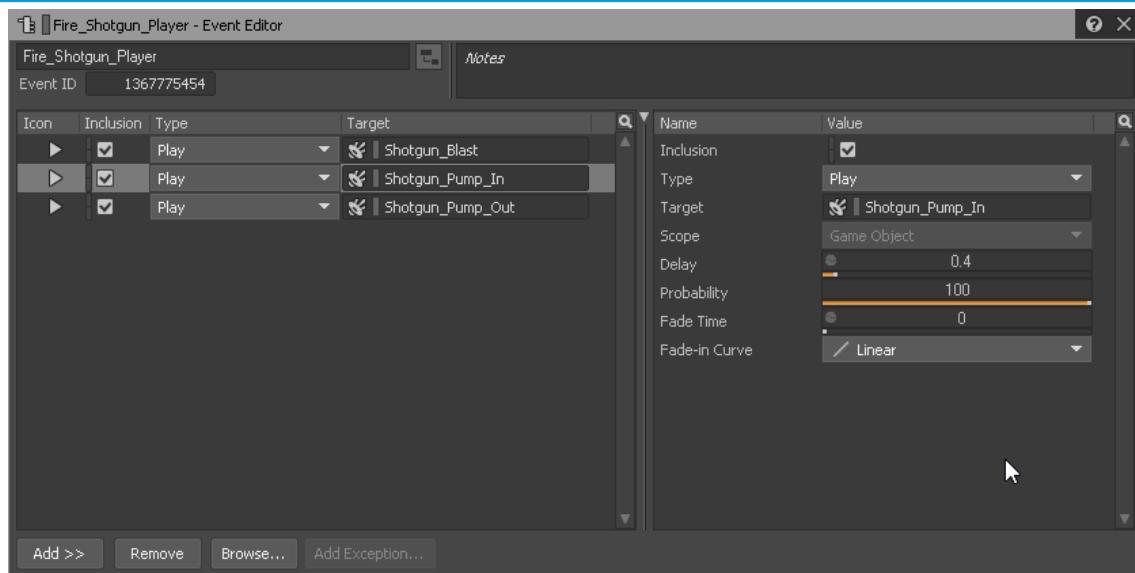
There are various properties associated with different types of actions, but you must choose to display those properties using the Event Editor's disclosure triangle.

6. In the upper right corner of the Event Editor, click the disclosure triangle.



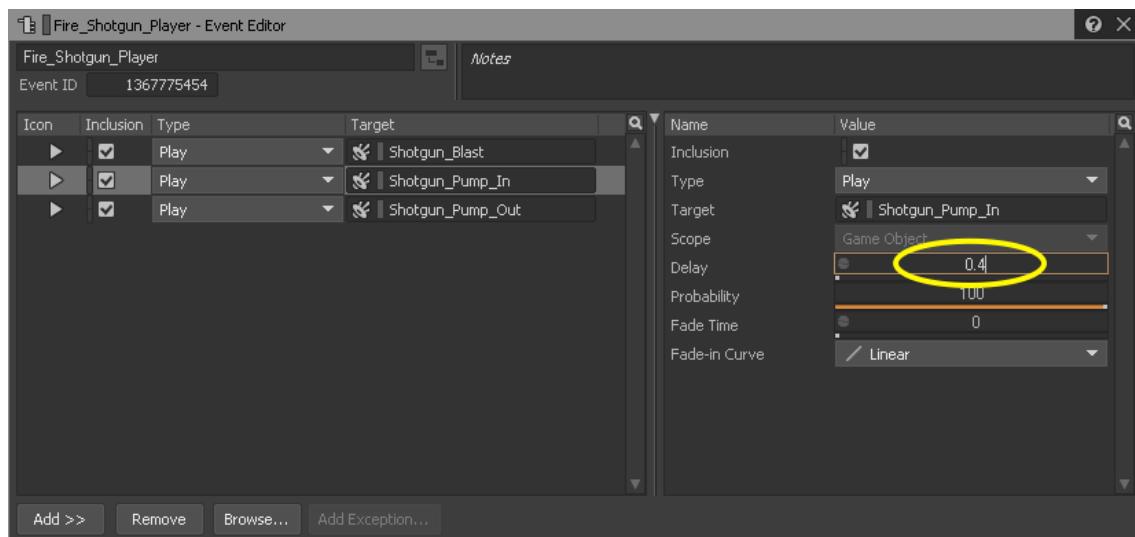
Once the disclosure triangle is clicked, you see all of the properties related to the selected action appear in a pane on the right side. The Play action type includes a Delay property.

Lesson 2: Designing a Soundscape



In the game, the visual animation of the player firing the shotgun shows the pump in action occurring about .4 seconds after the player fires the gun. Delay property for all of the actions in a column within the Actions pane on the left. This makes it much easier to understand the timing relationships between the actions.

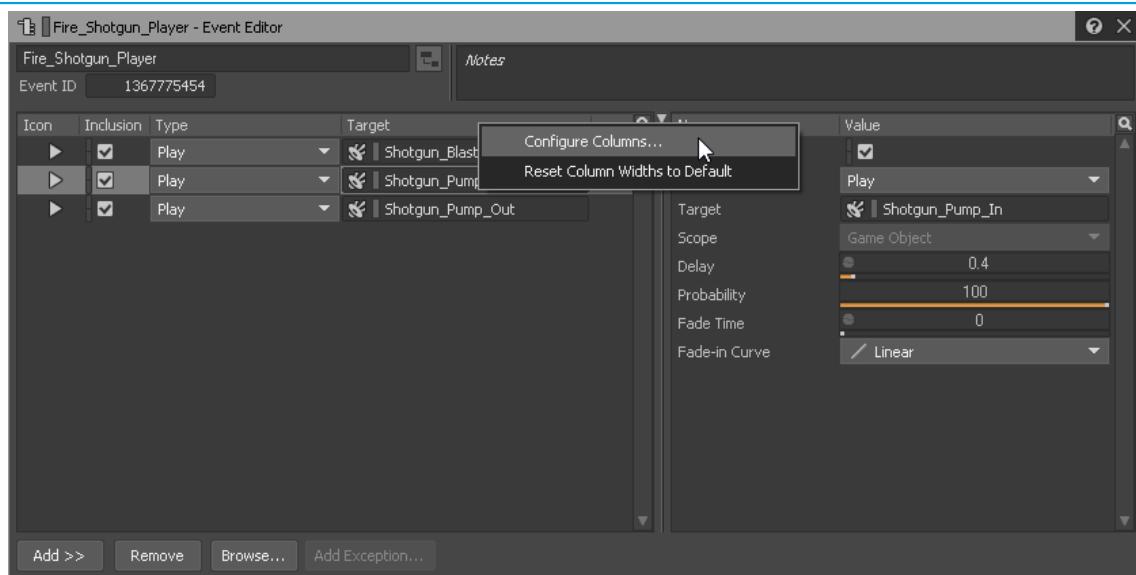
7. Click the Delay parameter for Shotgun_Pump_In, type 0.4 and press Enter.



You need to do the same for the pump out sound, but with even more delay. It would be helpful to see the delay property of all of actions simultaneously in order to more easily understand the order in which they'll play. You can add properties columns to the action list on the left Event Editor pane.

8. Right-click any of the column headers and choose **Configure Columns...**.

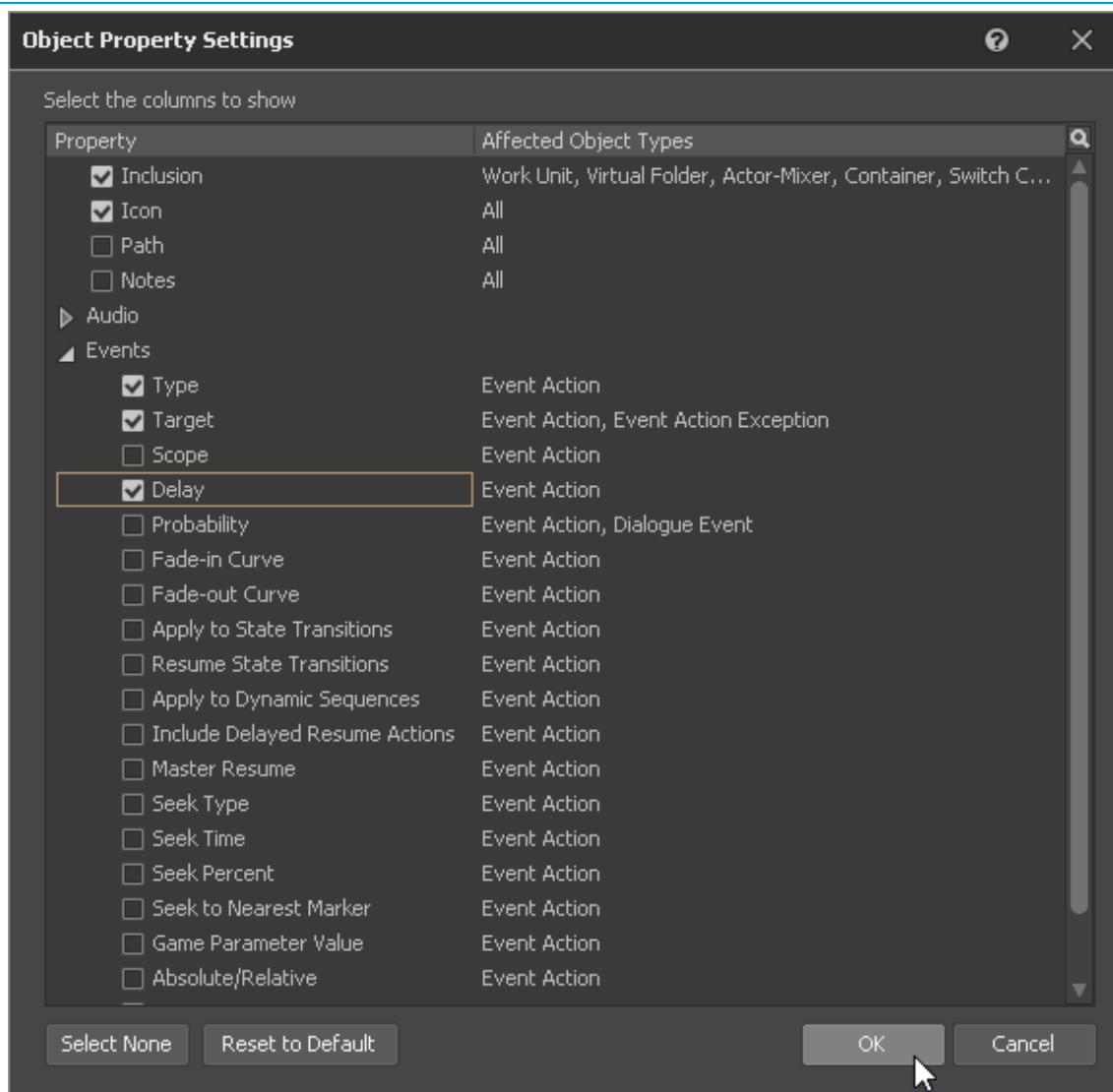
Lesson 2: Designing a Soundscape



An Object Property Settings window appears.

9. Expand the Events section, select Delay and click OK.

Lesson 2: Designing a Soundscape

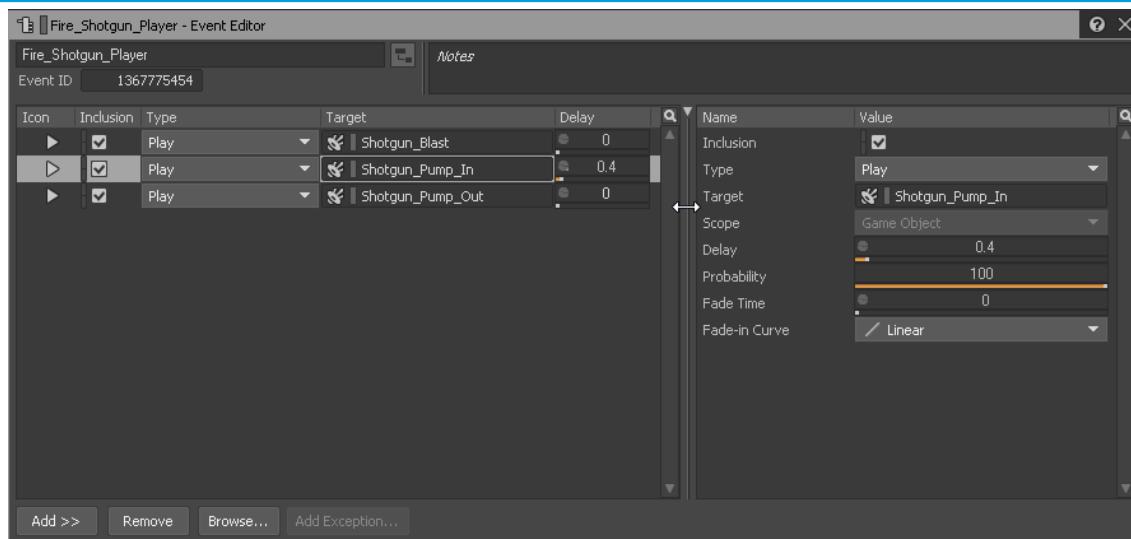


You now see the Delay property displayed for all of the actions.



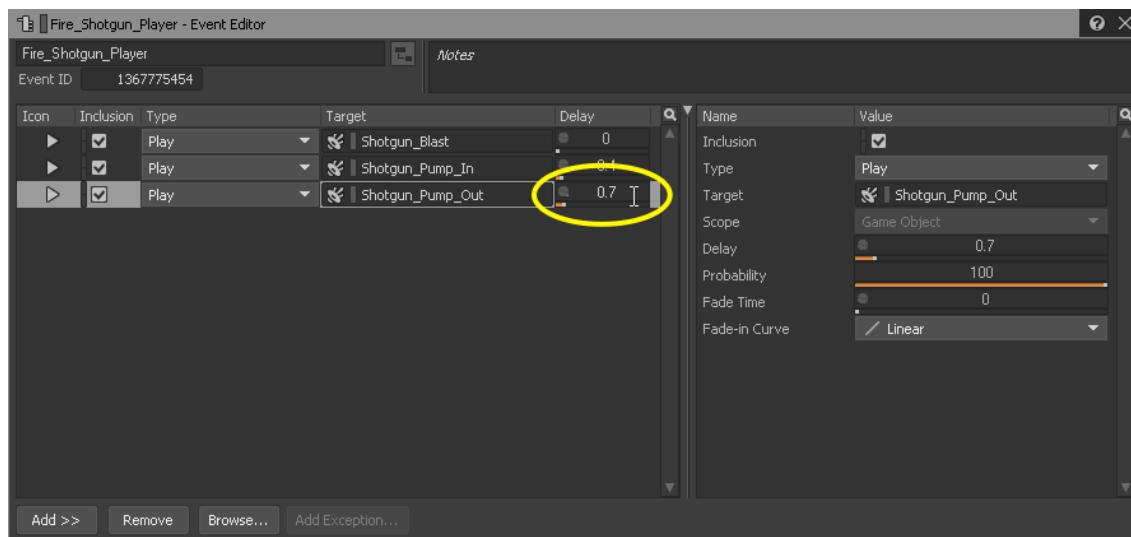
If you don't see the Delay column in the left Actions pane, then drag the border between the two panes in the Event Editor to the right until the Delay column is visible.

Lesson 2: Designing a Soundscape



The pump-out sound needs to be heard about a little more than a half second after the initial blast, so 0.7 would be a good choice.

10. Change the Shotgun_Pump_Out delay value to 0.7 and press Enter.



Even with the Delay property displayed as a column, they will not necessarily be displayed in chronological order. Click the Delay column header to sort into either ascending or descending order and then click again to reverse the order.

Now you can test the entire sequence of the blast with the shotgun pumps.

11. Play the Fire_Shotgun_Player event.

Using Randomization

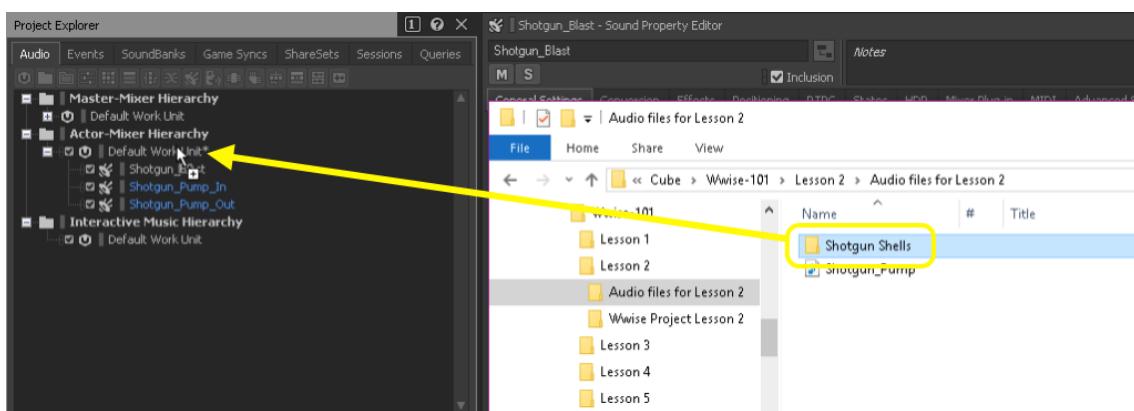
One of the most powerful aspects that audio brings to video games is how it can drive the experience of the player in ways that a visual alone cannot. Audio's role is not limited to merely supporting the graphics that you see on the screen, so it's important to think beyond what you see in the game. More importantly focus on what things that you don't see. For example, the purpose of pumping a shotgun is to expel the used shell so that another can be loaded. Where does the shell go? It falls on the ground of course, but there's nothing about the shotgun animation that shows this. This doesn't mean that you shouldn't put shells in. In fact, you definitely should! This is what helps to sell the experience to the player.

You'll now add the sound of shotgun shells hitting the ground. Every time the shells hit the ground, they bounce in a different pattern creating a different sonic rhythm, which means that using the same recording of shells bouncing on the ground over and over would quickly sound unnatural and become an audible nuisance for the person playing the game. To avoid this, you can import multiple audio files, each created from dropping actual shotgun shells on the ground. You'll then use a type of audio object called a Random Container to select one of these imported files each time the shotgun is fired.

Importing an Audio File Folder

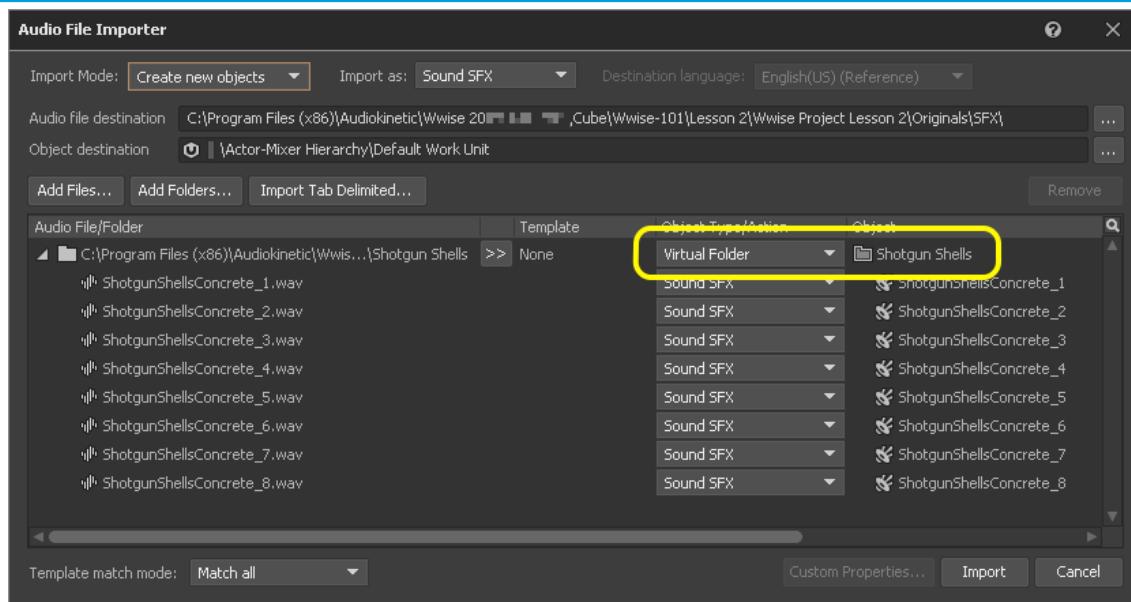
In the last lesson, you added the shotgun blast sound using an import process you initiated from within Wwise. You can also drag audio files, or even folders containing audio files from a system window directly into your Wwise project.

1. Within your operating system, navigate to the **Wwise-101 > Lesson 2 > Audio files for Lesson 2** and drag the **Shotgun Shells** folder onto the Actor-Mixer Hierarchy's Default Work Unit.



The Audio File Importer automatically opens.

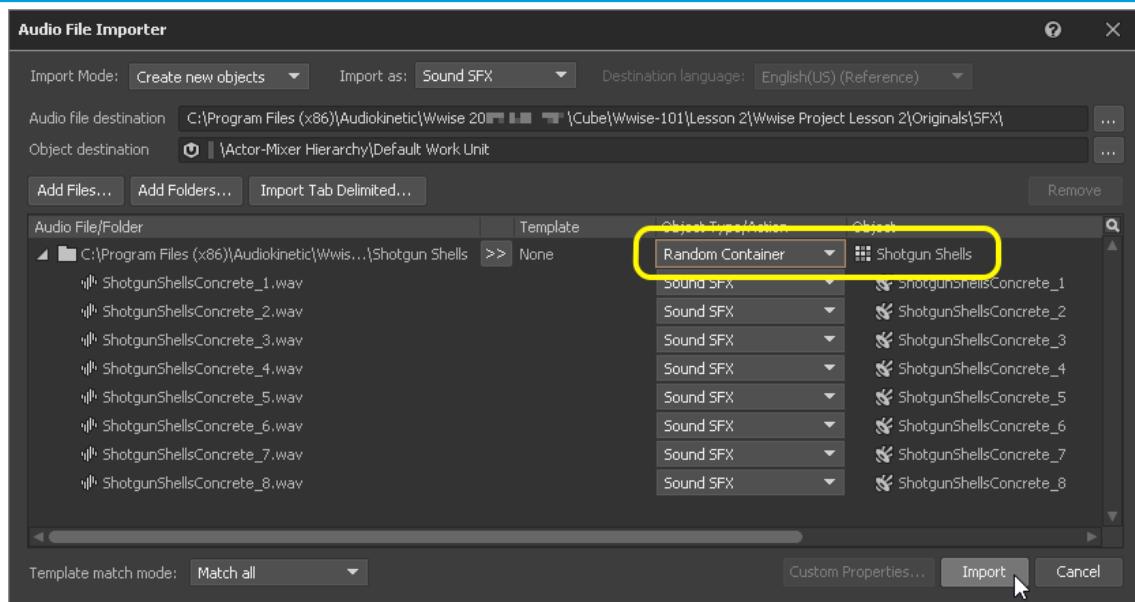
Lesson 2: Designing a Soundscape



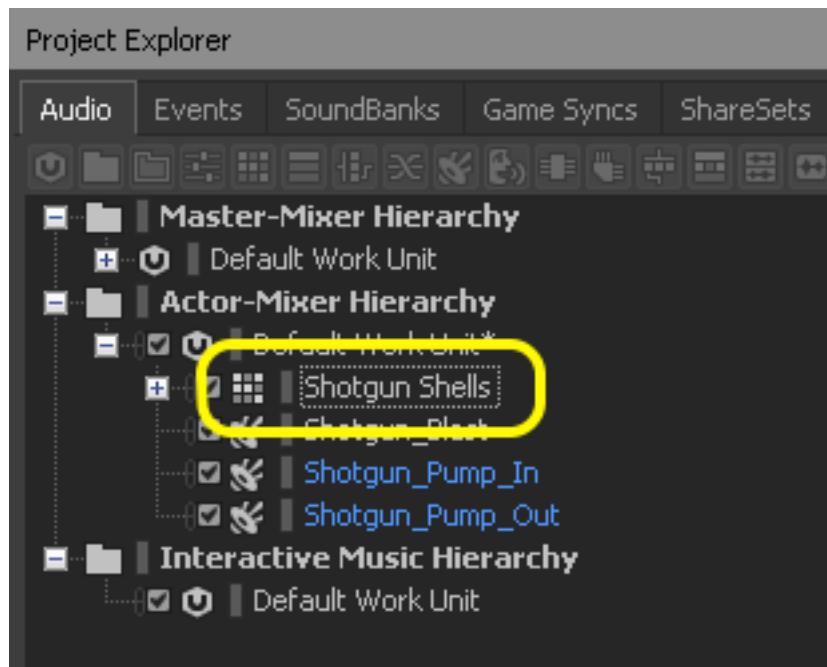
You see that the audio files contained within the folder you're about to import will be brought in as Sound SFX objects. The folder itself is represented in the top line as an object that will be imported into the hierarchical structure. By default the folder will be imported as a Virtual Folder, which is purely a way to organize objects within a Wwise Hierarchy. Instead you can change the object type to Random Container. Think of Random Container objects as a folder with some added functionality. Like a Sound SFX object, Random Containers can be selected and played using the play button in the Transport Control, in which case, one of the objects contained within will be randomly selected and played. This is a perfect way to add variety to sounds that would otherwise be annoyingly repetitive if the same audio file were played every time.

2. In the first line, change the Object Type/Action to Random Container and then click Import.

Lesson 2: Designing a Soundscape

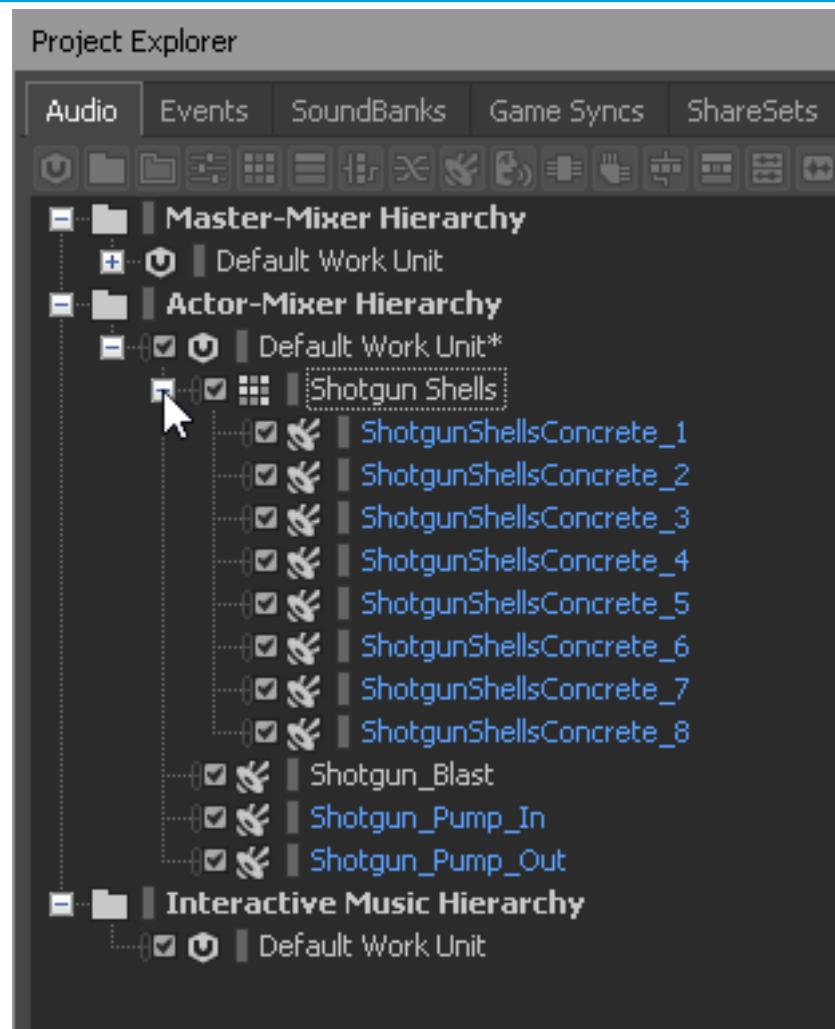


In the Project Explorer, you now see a new object named Shotgun Shells that has a different icon than the Sound SFX object you've used to this point.



3. Expand the Shotgun Shells Random Container object.

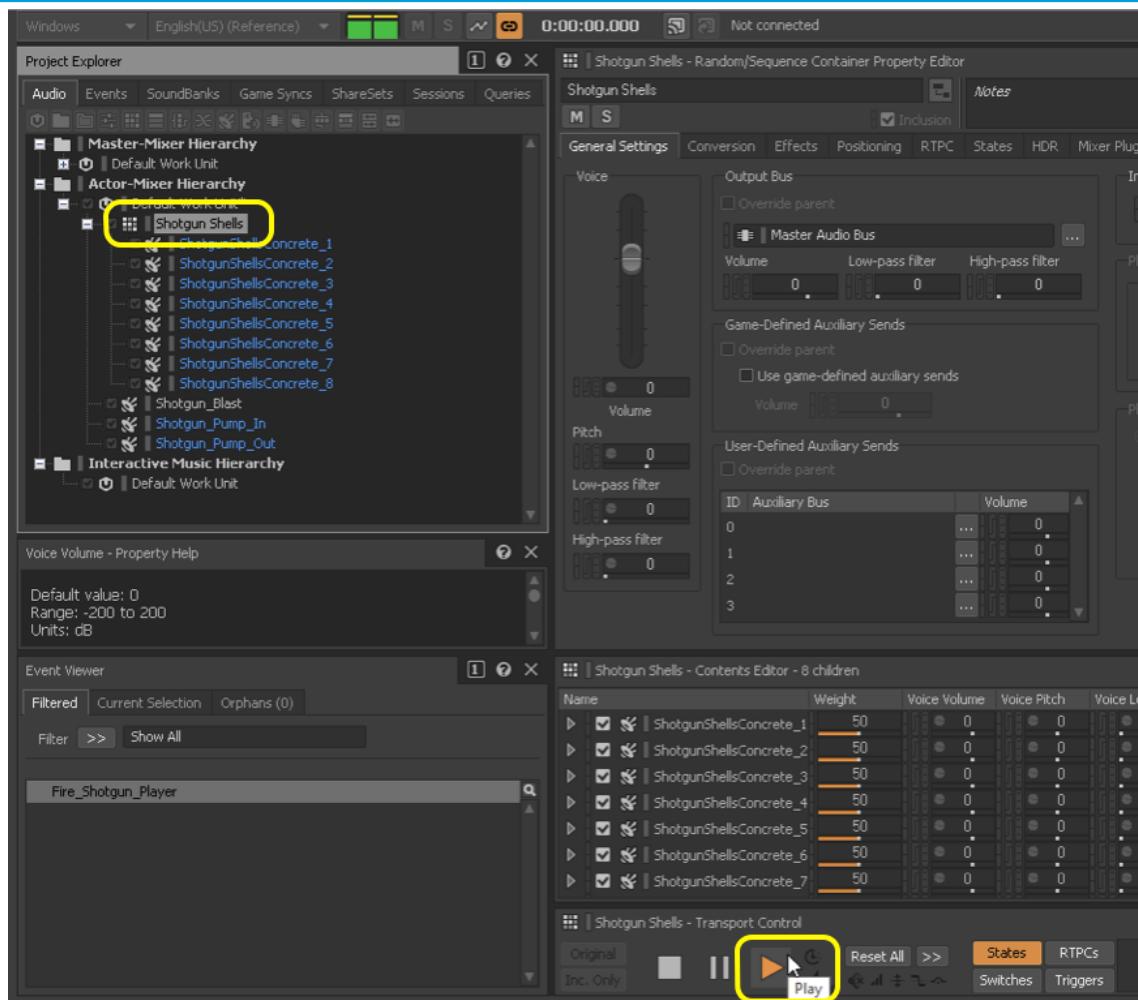
Lesson 2: Designing a Soundscape



You can see that all of the shotgun shell recordings have been imported as Sound SFX objects and they are contained within the Shotgun Shells Random Container.

4. Select the Shotgun Shells Random Container and play the object multiple times.

Lesson 2: Designing a Soundscape



Notice that the sound of the shells dropping on the ground varies because each time you play the Shotgun Shells object, Wwise is randomly choosing one of the contained shotgun shell recordings.

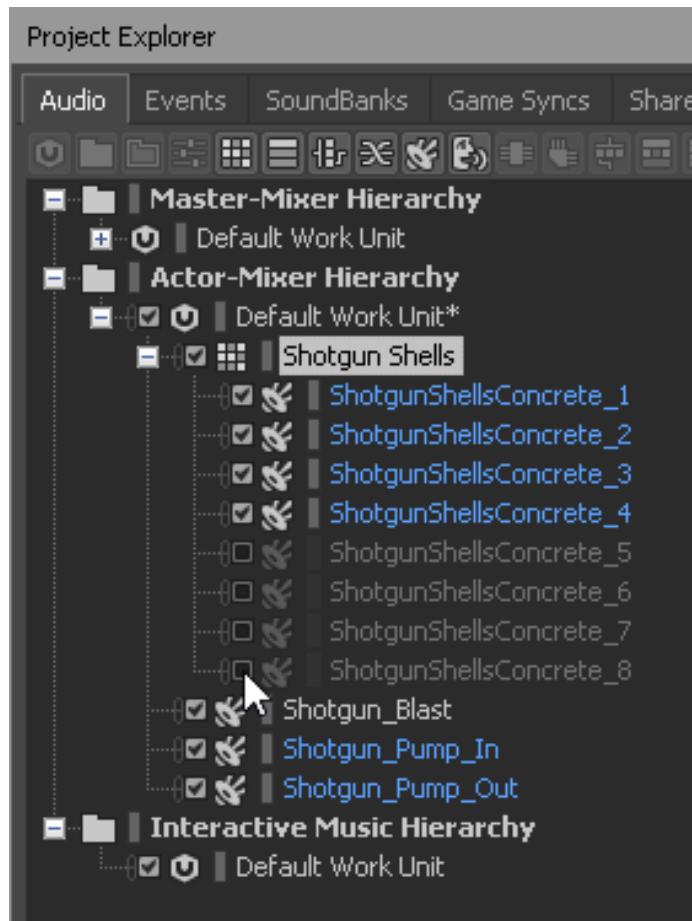
You may find that some of the shell sounds don't suit you, or you may have concerns about having too many sounds which uses more memory at run-time. You can choose to include or exclude specific shell recordings without having to permanently delete them from the Wwise project. This is accomplished using the platform inclusion/exclusion check boxes that appear to the left of every object in the Project Explorer. You can think of these check boxes somewhat like a mute button on a mixing console, as de-selecting the check box removes its role in the creation of the soundscape, but provides a quick way to add it back in if you later decide you want it.



You'll learn more about how the platform inclusion/exclusion check boxes can be used for optimizing a project in Lesson 7.

Lesson 2: Designing a Soundscape

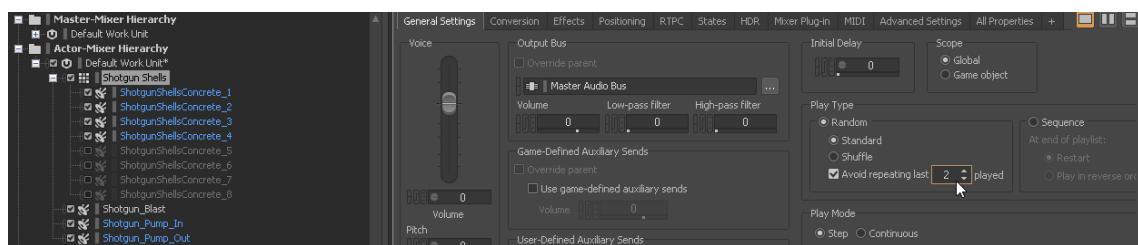
5. De-select the platform inclusion/exclusion check boxes for the last 4 shells.



Using Controlled Randomization

While the shotgun shells are randomized, it's still possible that the same shell sound could be randomly selected multiple times in a row. This can be avoided by adjusting properties that let you control how randomization occurs.

1. In the Shotgun Shells Property Editor, confirm that the **Avoid repeating last** check box is selected and change the value to the last 2 played.



Now it will be impossible to hear the same exact shell sound played multiple times in a row.

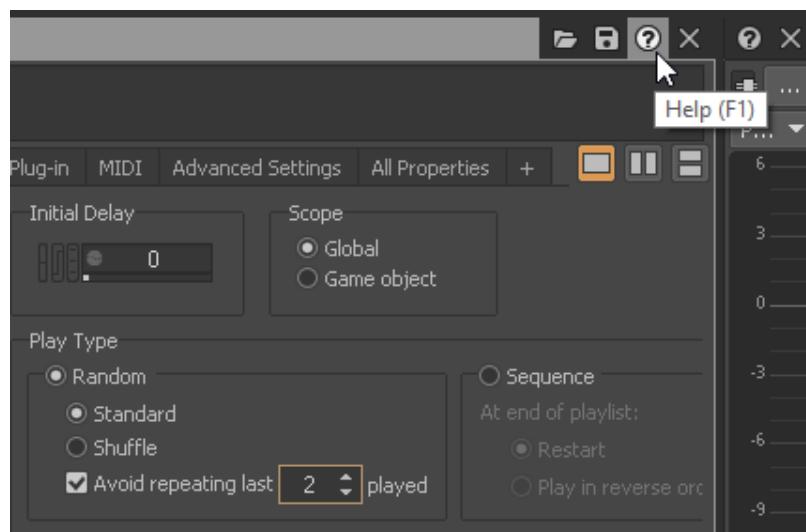
Lesson 2: Designing a Soundscape

You may notice the Standard and Shuffle randomization radio buttons and wonder how they differ. Remember that you have the Property Help view that you used earlier that will explain each properties functionality when selected. In addition, you can easily view the documentation for all of the properties found within view by click Help button displayed as a "?" in the upper right corner.



By default, Wwise uses an online help system that requires an Internet connection. If you know you might be working without an Internet connection, you can choose to install an offline set of documentation from the Wwise Launcher. You must indicate that you want to use the offline documentation by going to the main menu and choose Help > Documentation Source > CHM.

2. Click the help button in the upper right corner of the Property Editor view.



A web browser page will open to the Wwise online documentation that provides information about all of the properties currently displayed in the Property Editor.

Lesson 2: Designing a Soundscape

The screenshot shows the Wwise Help website interface. The top navigation bar includes links for SEARCH, ENGLISH, GET Wwise, and LOG IN. The main content area is titled "Wwise Help" and "Property Editor: Random/Sequence Container". The left sidebar contains a tree view of Wwise Reference topics, including Project, Views, Layouts, Edit, Audio, Windows, and Help, Project Explorer, and various Property Editors for different objects like Random/Sequence Container, Property Editor: Blend Container, and Common Property Table.

Property Editor: Random/Sequence Container

This Property Editor contains the properties and behavior options for the selected Random or Sequence Container. Object properties determine the characteristics of the sound and motion objects when they are played in-game. Object behaviors determine which sound and motion objects are played back at any given point in the game.

A Random Container is a group of one or more objects that are played back in a random order. A Sequence Container, on the other hand, is a group of one or more objects that are played back according to a specific playlist.

Note

An object, in this case, includes containers.

The General Settings tab of the Property Editor is divided into three separate areas. The first area includes all the absolute properties, such as output routing. The middle area includes all the relative properties, such as volume and pitch. The last area, on the right, includes all the behaviors. For a complete description of absolute and relative properties, refer to [About Properties in the Project Hierarchy](#).

For a description of the properties on the Conversion, Effects, Positioning, RTPC, States, and Advanced Settings tabs, refer to [Common Property Tabs: Actor-Mixer Objects](#).

Note

The properties and behaviors for the Random and Sequence Containers are displayed in the same Property Editor.

Refer to [Understanding the Voice Pipeline](#) to learn about how voices are being processed, how they are being routed and where the different volumes and Effects are being applied.

General	Description
Inclusion	Determines whether the element is included or excluded. When selected, the element is included. When unselected, the element is not included. By default, this applies across all platforms. Use the Link Indicator (to the left of the check box) to determine or to set platform specific customizations.
Name	The name of the object.
Notes	Any additional information about the object properties.
	Controls the Mute and Solo states for the object and shows the implicit mute and solo states for the

At the top there's an article describing the difference between the Random Container object you just created, and a Sequence Container that you'll create later in this lesson. Scrolling down, you'll see all of the properties for the current view along with the same information that is available in the Property Help view.

3. Scroll down the web page until you reach the Random type property and review the difference https://wwwaudiokinetic.com/library/?source=Help&id=creating_random_container#using_random_containers_examplebetween Standard and Shuffle.

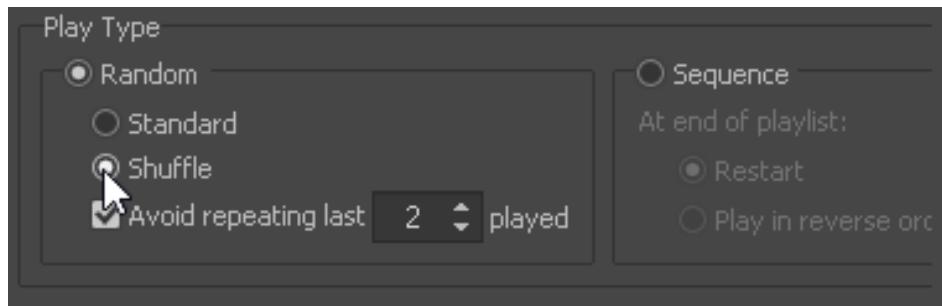
Play Type

Random or Sequence	<ul style="list-style-type: none"> Random: Defines the container as a Random Container, which means that objects within the container will be played back in a random order. There are two different modes of random playback: Standard and Shuffle. Sequence: Defines the container as a Sequence Container, which means that objects within the container will be played according to a user-specified playlist. <p>Default value: Random</p>
	<ul style="list-style-type: none"> Standard: Keeps the pool of objects within the container intact. After an object is played, it is not removed from the possible list of objects that can be played and can therefore be repeated. Shuffle: Removes objects from the pool after they have been played. This option avoids repetition of sounds until all objects have been played. The last object played cannot be repeated when the list is reset. <p>Default value: Standard</p>
Avoid repeating last x played	<p>Determines how many other objects must be played before an object can be repeated. The behavior of this option is affected by whether you are in Standard or Shuffle mode. In Standard mode, the object played is selected completely randomly, but the last x objects played are excluded from the list. In Shuffle mode, when the container is looped, the last x objects played will be excluded from the list. If you have a non-looping Random Container that is set to shuffle and continuous, the "Avoid repeating last x played" option has no effect on the playback of your container.</p> <p>Default value: 1 Range: 1 to 999</p>

Lesson 2: Designing a Soundscape

You can see that Shuffle provides even more protection from hearing the same shotgun shells played closely together.

4. Close the web browser and go back to Wwise. Then click the Shuffle radio button and play the Shotgun Shells Random Container to hear the effect.



Randomizing Properties

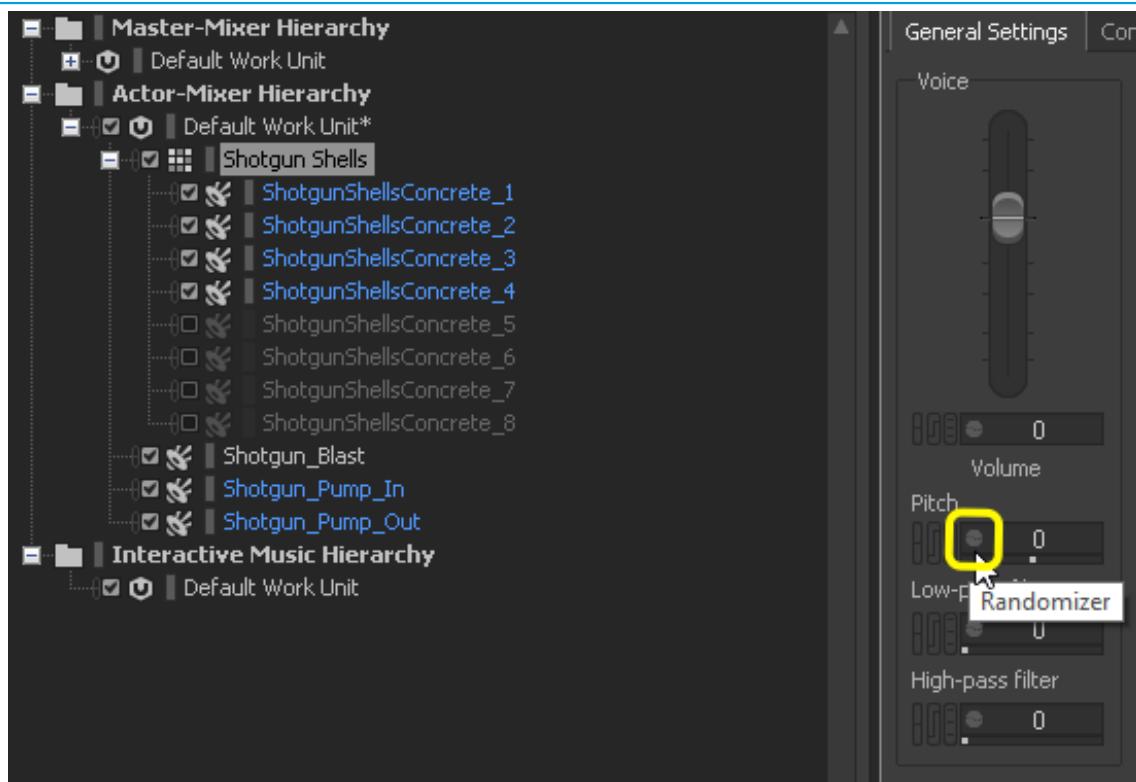
Beyond randomizing sound selection, in Wwise you can randomize the value of nearly every property that can affect your sounds, such as volume, filters, and pitch. This, in combination with randomizing sound selection, makes for an even greater variety of sound, without adding to the amount of memory needed at run-time.

The Randomizer button is found to the left of a properties value and looks like a small gray circle with a squiggly line through it.

You'll further modify the shotgun shells by slightly randomizing the pitch of the shotgun shell recordings.

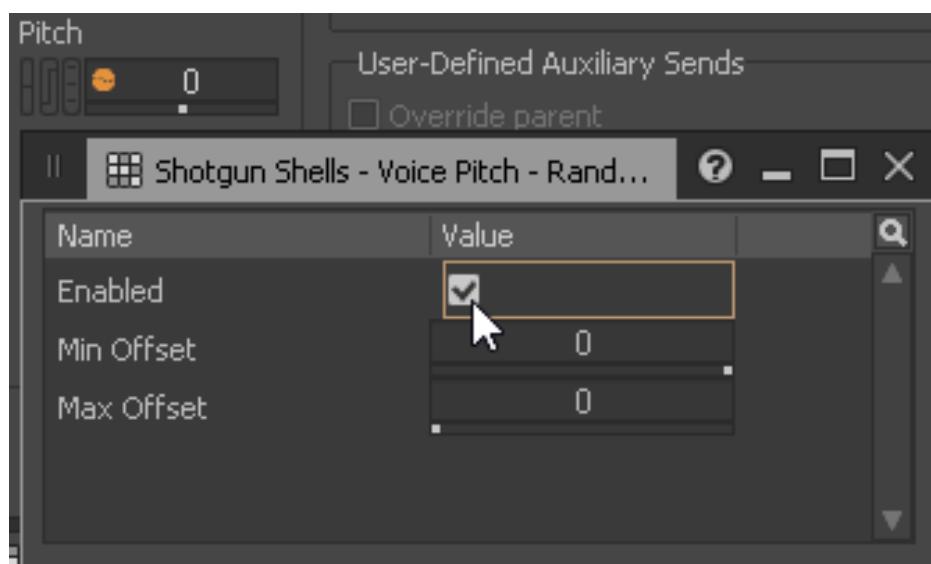
1. In the Shotgun Shells Property Editor, double-click the Randomizer icon for the Pitch property.

Lesson 2: Designing a Soundscape



The Randomizer window opens, displaying an Enable check box as well as settings for minimum and maximum values. To hear any change that you're about to make, you'll need to first enable randomization for the pitch property.

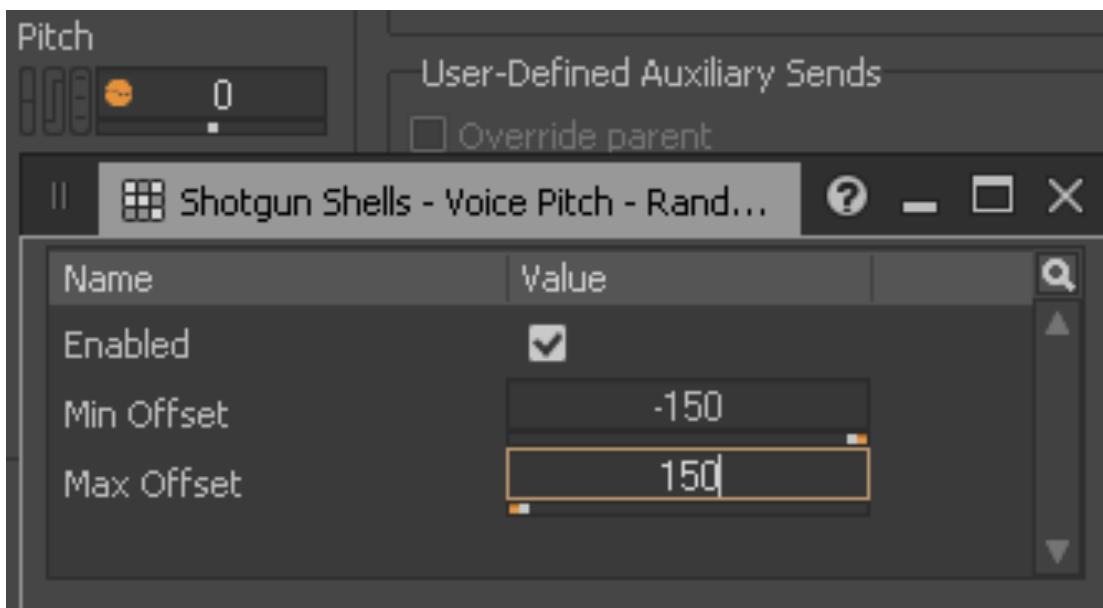
2. Click the **Enable** check box.



Lesson 2: Designing a Soundscape

Notice that the Randomizer icon for the Pitch property has turned yellow, indicating the randomization is now active. Although randomization is technically activated, you won't hear any difference until you modify the Min/Max values. These Min/Max values are the range of relative offset values that center around the value set in the UI for the property. In the case of pitch, the value is expressed as cents. There are 100 cents of pitch per musical half step and a total of 1200 cents per octave.

3. Enter -150 and 150 into the Min and Max fields.



Now, every time a shotgun shell is played, the pitch will be anywhere between 150 cents below or above the pitch that the sound was recorded at.

4. Play the Shotgun Shells multiple times and notice the variation in the pitch each time it's played, then close the Pitch Randomizer window.

Related Video

[Wwise-101-03 - Using Randomization](#)

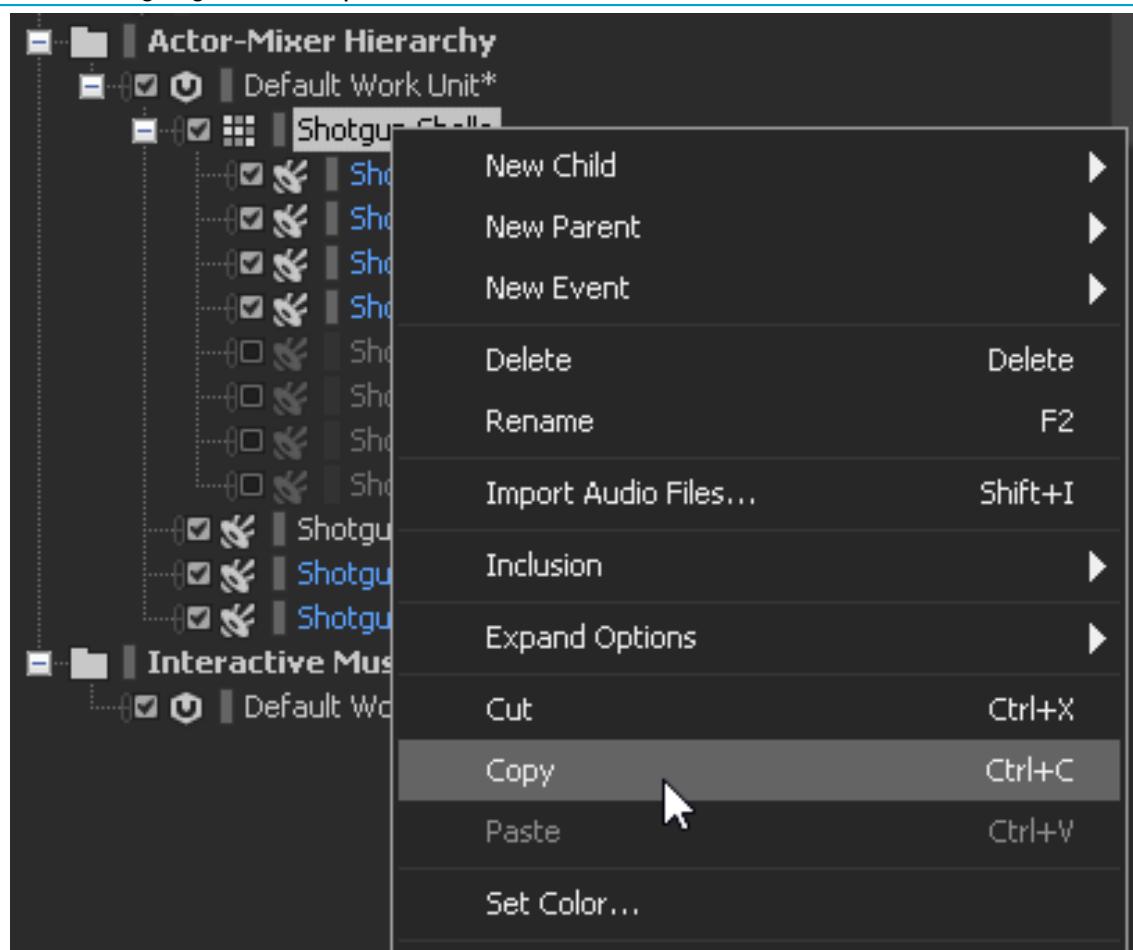
Granulate Your Sounds

Now you're going to provide even more differentiation in the shotgun shell sounds by dividing the shell recordings into two parts; the initial strike of the shell hitting the ground-head, followed by the sound of the shell bouncing across the floor-tail. By dividing the sound into multiple parts, you'll be able to randomly combine different head and tail sounds creating even more overall variety.

You're going to start by creating a copy of the work you've already done.

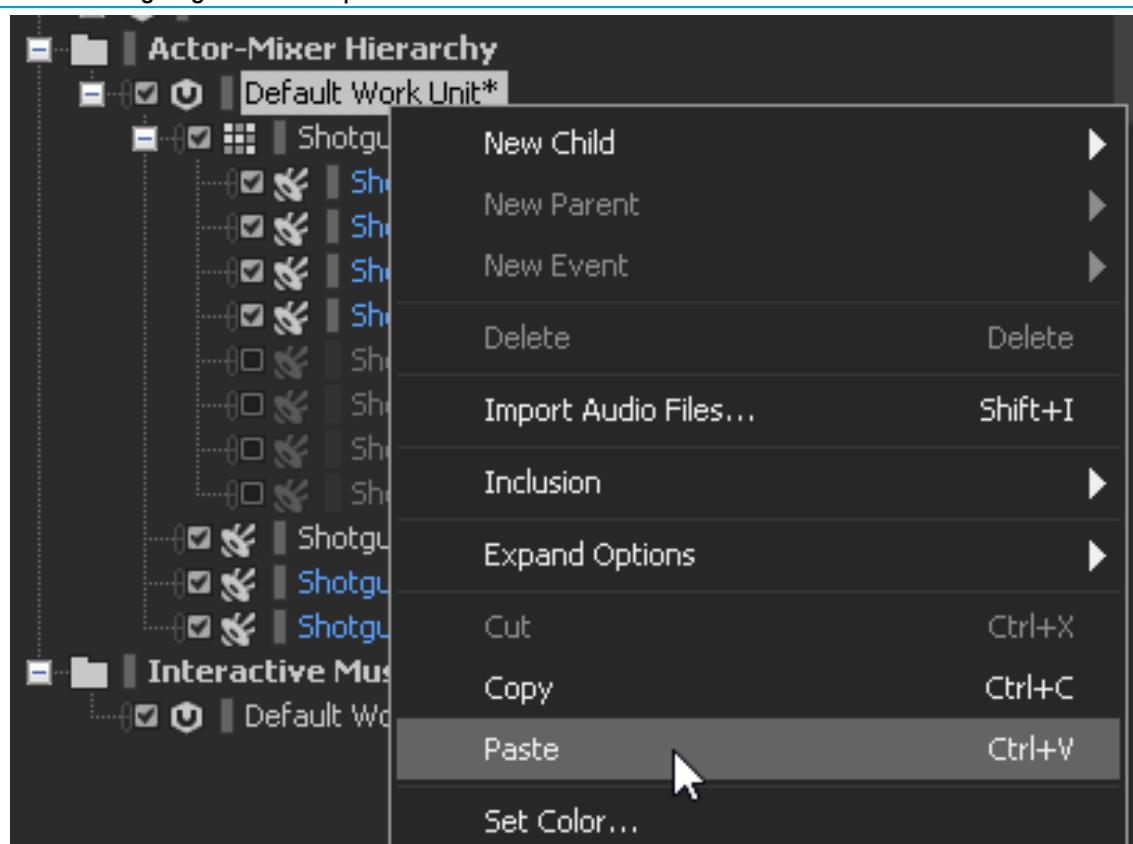
1. Right-click the Shotgun Shells object and choose **Copy**.

Lesson 2: Designing a Soundscape



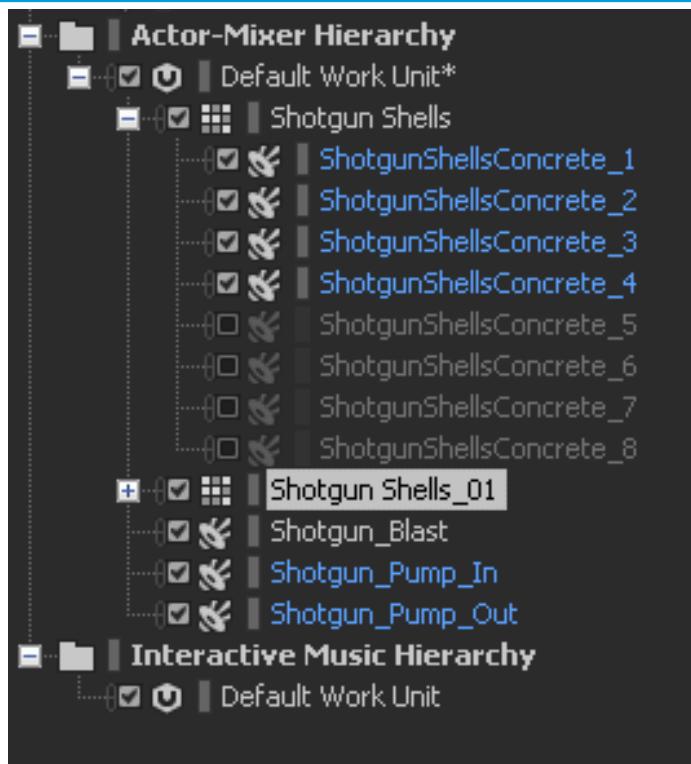
2. In the Actor-Mixer Hierarchy, right-click the Default Work Unit and choose Paste.

Lesson 2: Designing a Soundscape



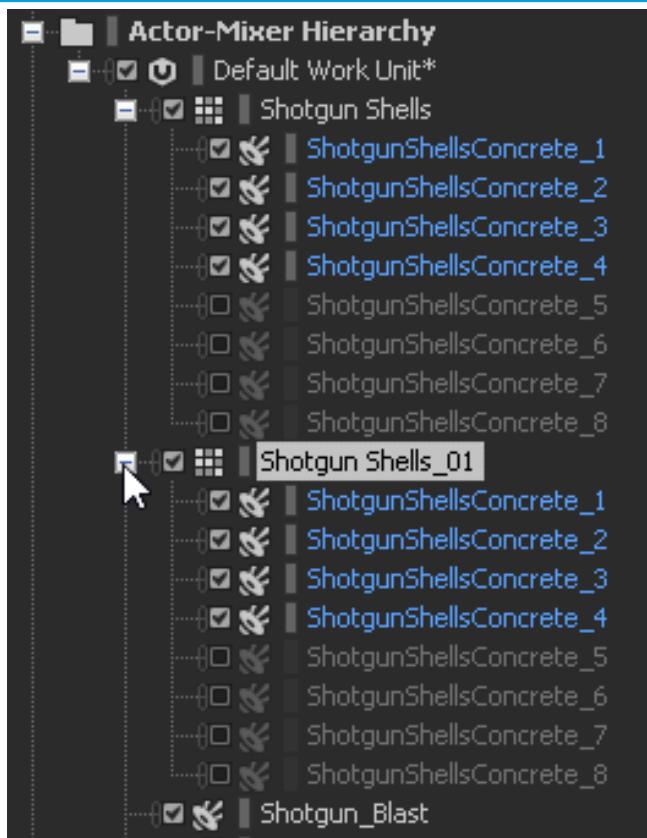
A duplicate Random Container called Shotgun Shell_01 is created.

Lesson 2: Designing a Soundscape



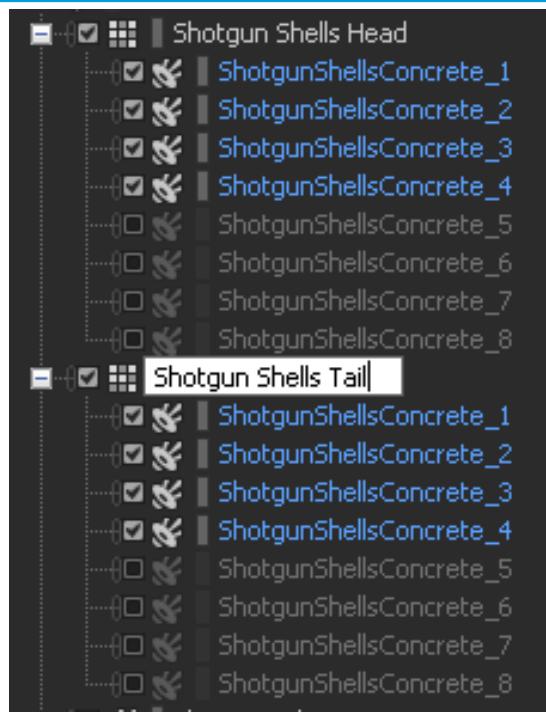
3. Expand the Shotgun Shells_01 to display its contents.

Lesson 2: Designing a Soundscape



4. Rename the two Shotgun Shells Random Container objects Shotgun Shells Head and Shotgun Shells Tail respectively.

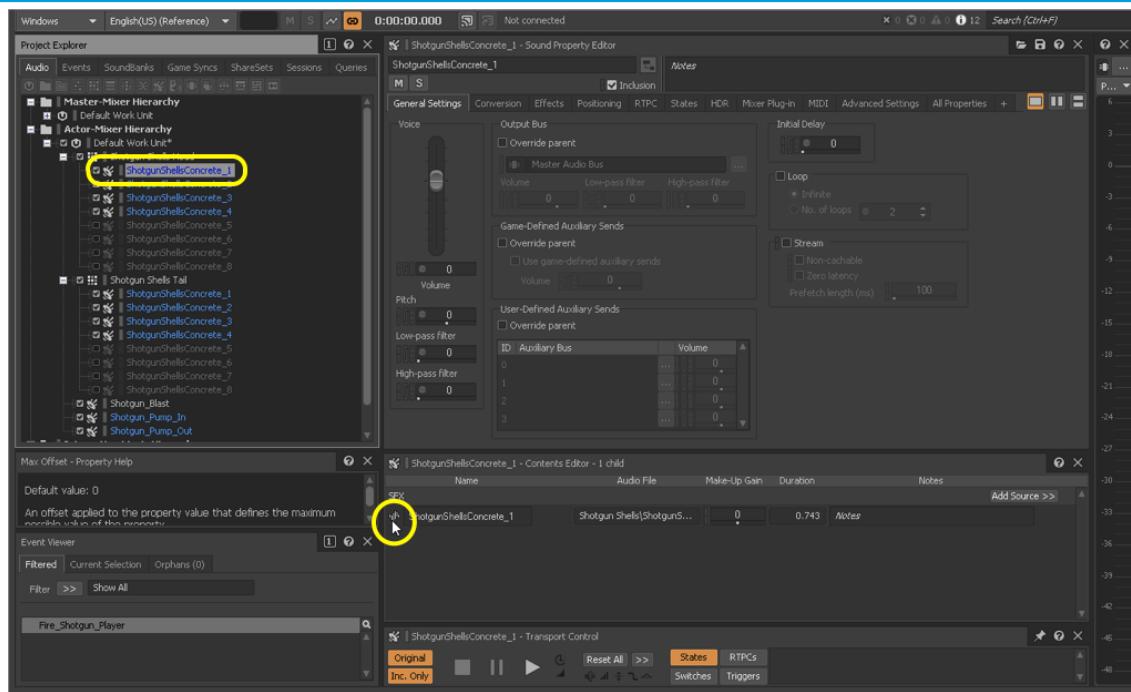
Lesson 2: Designing a Soundscape



You've effectively created two Random Containers that currently both have the exact same content. You'll now modify Sound SFX Objects contained within so they only play a certain section of the audio file that was imported. To do this, you'll need to view the waveform of the actual audio file contained within the Sound SFX Objects.

5. In the Shotgun Shells Head Random Container, choose the `ShotgunShellsConcrete_1` object and then in the Contents Editor double-click the `ShotgunShellsConcrete_1` source object icon.

Lesson 2: Designing a Soundscape

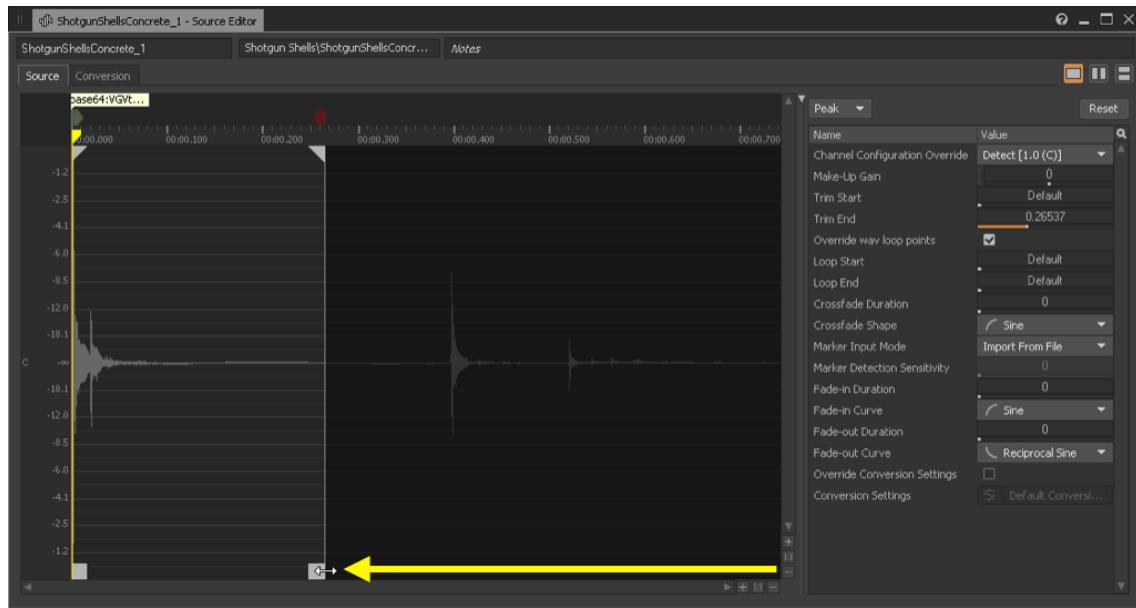


The Source Editor appears, displaying the audio file that's being used as a waveform. You can see the transients that show each time the shotgun shell bounced on the ground.

At the bottom of the waveform display are squares that designate the area of the audio file that will be played. These can be moved to play only a defined section of the audio file. For all of the shells in the Head Random Container, you only want to play the initial strike of the shell on the ground.

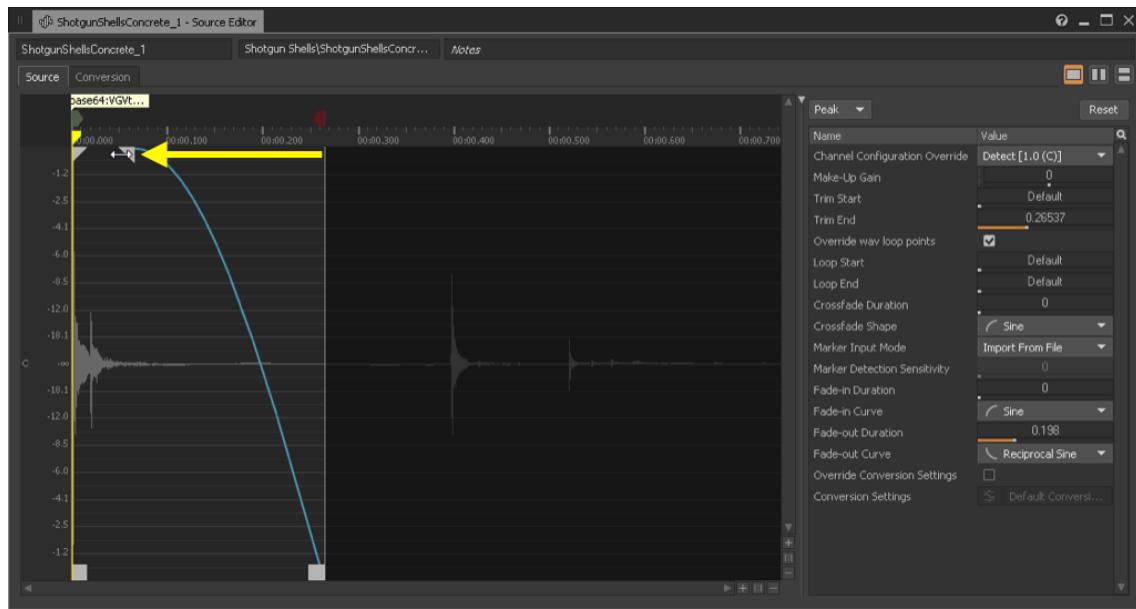
6. Drag the End position square so that only the first shell drop is selected and play the object to check your work.

Lesson 2: Designing a Soundscape



Sometimes clicks or pops can be heard when the sound is played, or sometimes you want a sound to gradually fade in or out. Getting rid of clicks or pops or providing fade-ins and fade-outs can be accomplished adjusting the triangles above the waveform display.

7. Drag the right fade triangle to the left to smooth the ending transition. Play the file repeatedly while making this adjustment and set it to your liking.

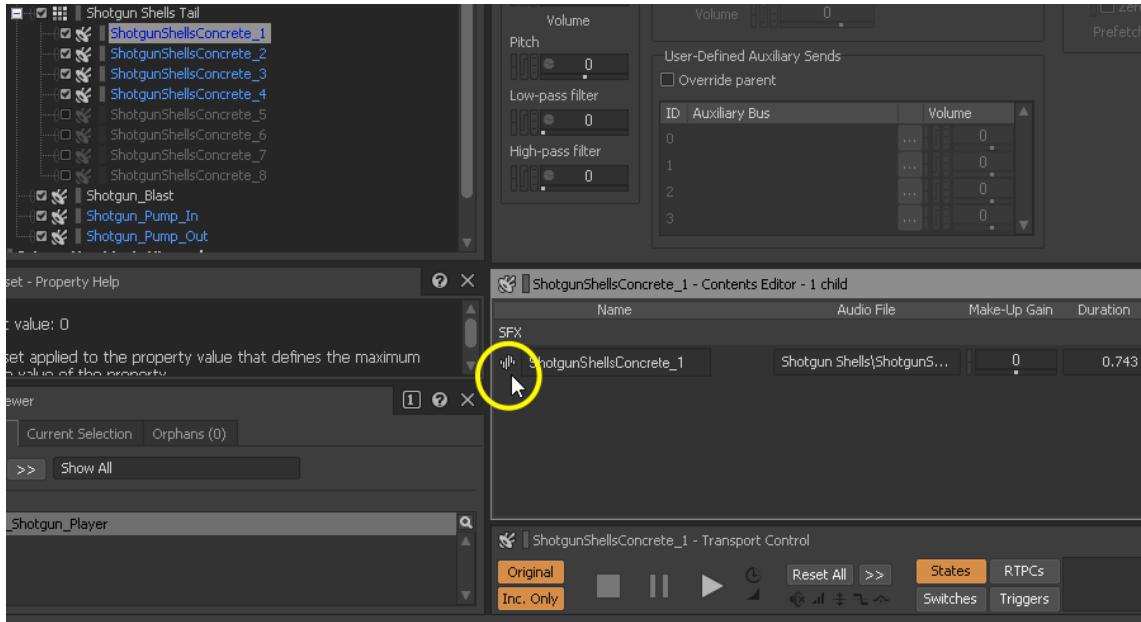


8. Repeat steps 5-7 on all the remaining objects within the Shotgun Shells Head object.

Lesson 2: Designing a Soundscape

Now you'll do a similar operation in the 'Shotgun Shells Tail' Random Container.

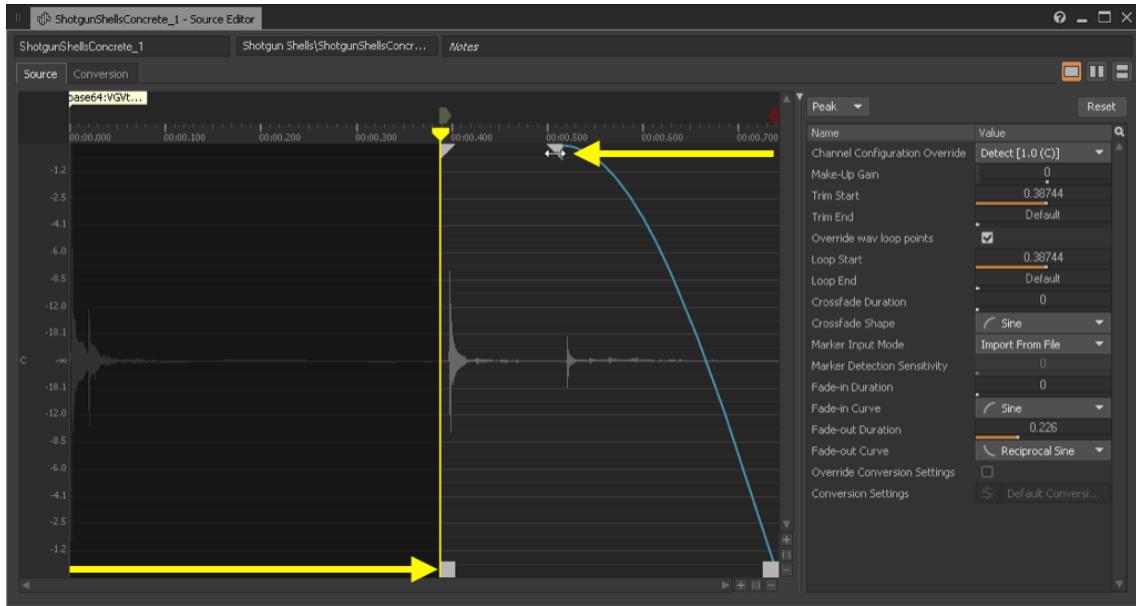
9. Expand the 'Shotgun Shells Tail' Random Container and select the ShotgunShellsConcrete_1 object.



For all of the Shell Tail Sound SFX objects, you're going to make adjustments in the Source Editor so that the initial strike is not heard but the sound of the shell bouncing on the floor after that initial strike is heard.

10. Adjust the start position of every Sound SFX object within the Shotgun Shells Tail Random Container so that the sounds play just before the latter transients. Make any necessary fades to smooth the sound.

Lesson 2: Designing a Soundscape



Creating a Sequence

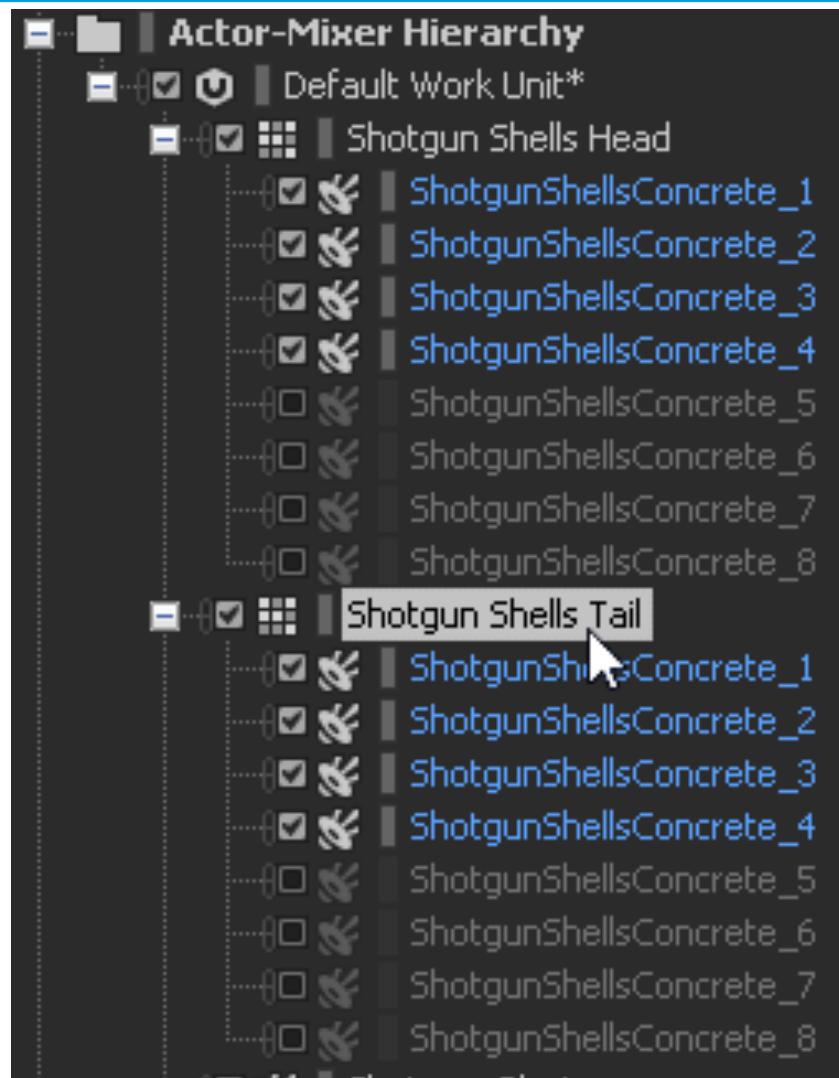
Now that you have your separate head and tail sections, it's time to tie them together so that each time a shotgun is fired, you first hear a randomly selected head, followed by a randomly selected tail.

You'll tie the head and tail together using another type of object called a Sequence Container. Similar to a Random Container, it holds other objects, but instead of randomly playing them, you can define a pre-ordered sequence of objects to play. In this case you'll set a sequence to play the head shell sounds first, followed by the tail shell sounds.

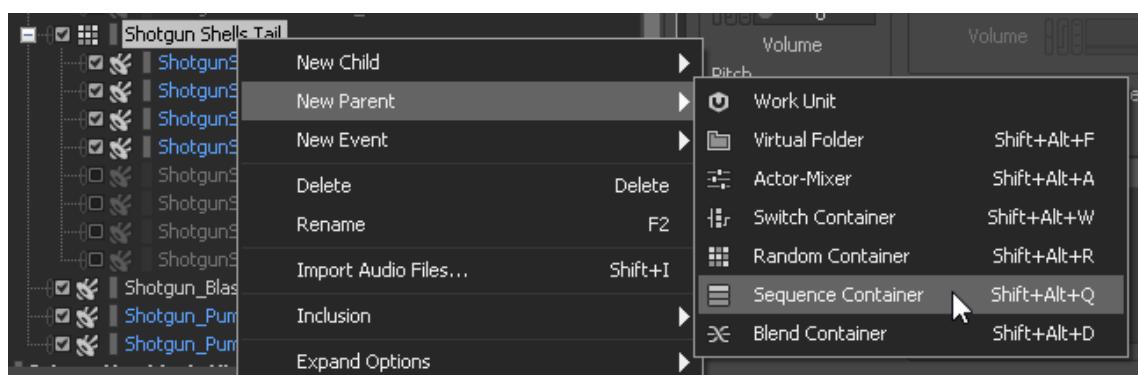
With randomization and the already randomized pitch in each head and tail Random Container, the listener will not likely realize that so few shell recordings were used.

You need to start by putting the Shotgun Shells Head and Tail random containers into a Sequence Container object.

1. Press and hold the **Ctrl** key and select the Shotgun Shells Head and Shotgun Shells Tail Random Containers.

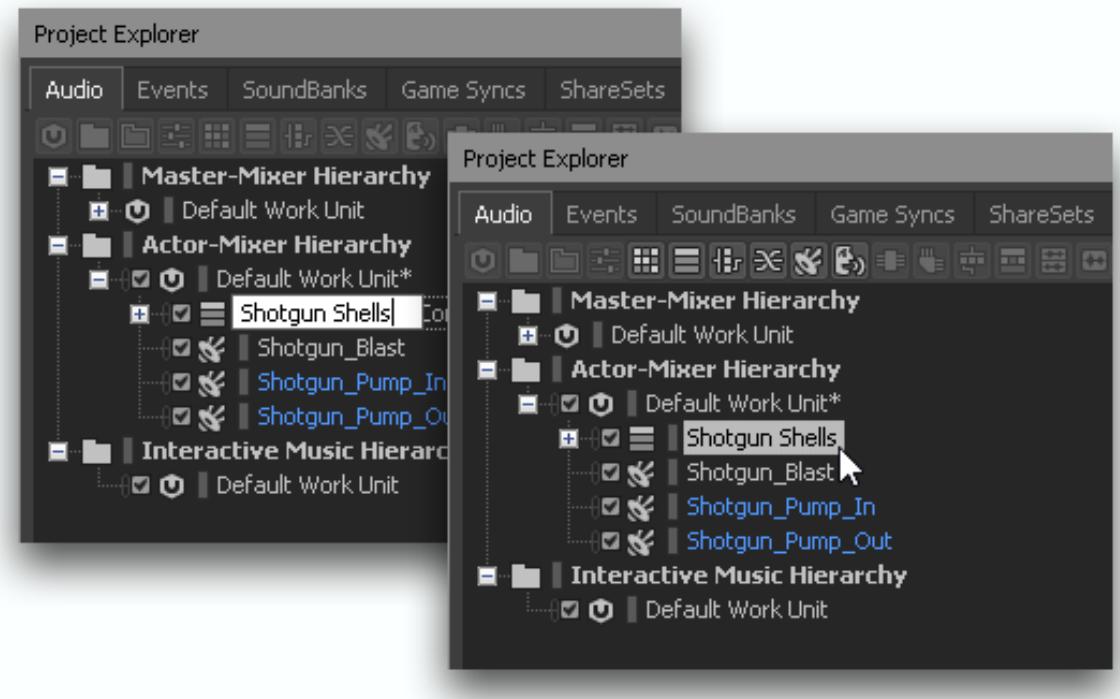


2. Right-click either of the two selected objects and choose New Parent > Sequence Container.



3. Rename the newly created object Shotgun Shells.

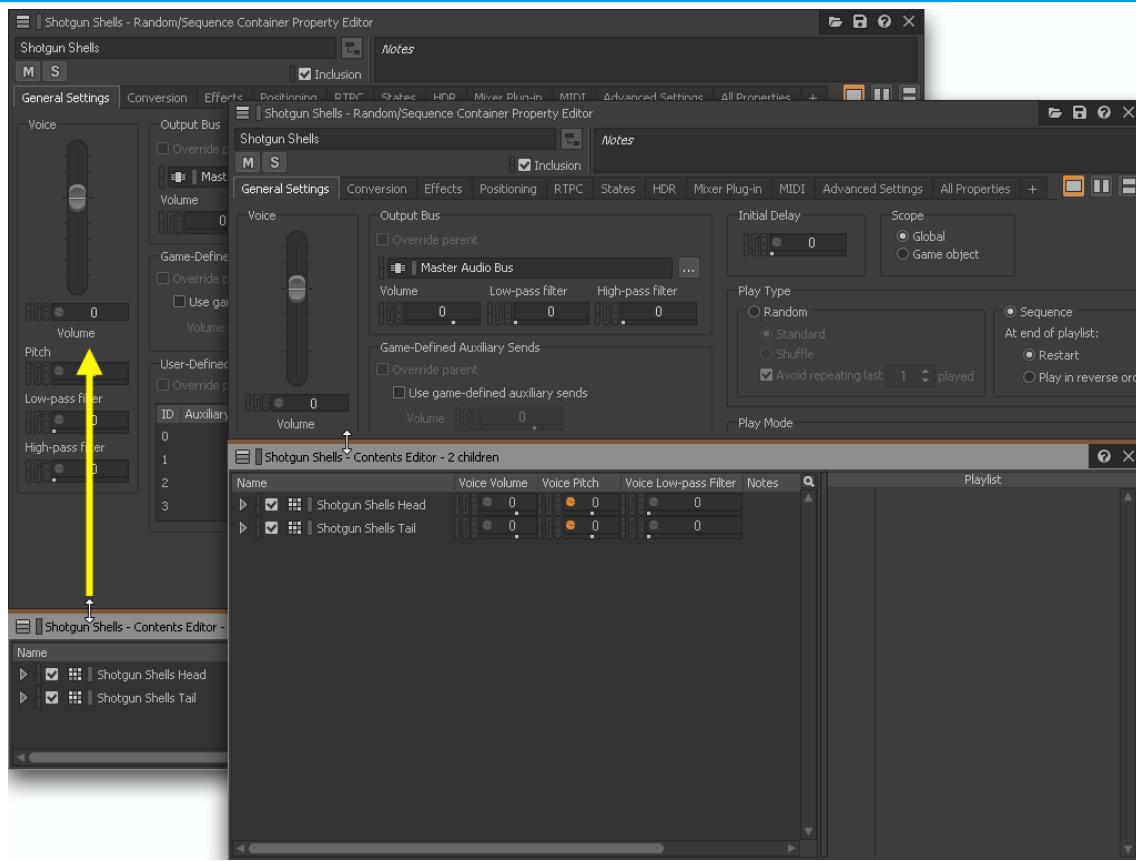
Lesson 2: Designing a Soundscape



Now you need to define the order that you want the contained objects to play in. You may need a little more room to work, so re-size the Shotgun Shells Contents Editor area.

4. Resize the Contents Editor to show more space.

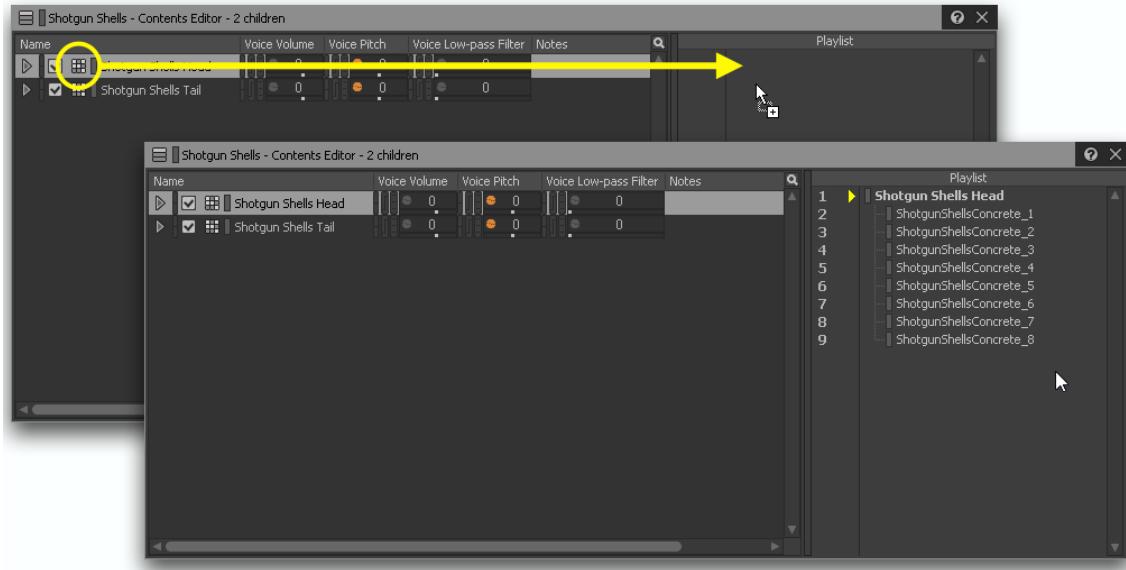
Lesson 2: Designing a Soundscape



On the left side of the Contents Editor, there are two Random Containers shown within the Shotgun Shells sequence container. The order in which they are listed does not reflect the order in which they will be played. The order is controlled by the Playlist, which is displayed as a column on the right side of the Contents Editor. You add items to the playlist by dragging them from available objects in the list from the left column into the Playlist in the right column.

5. Drag the Shotgun Shells Head into the Playlist.

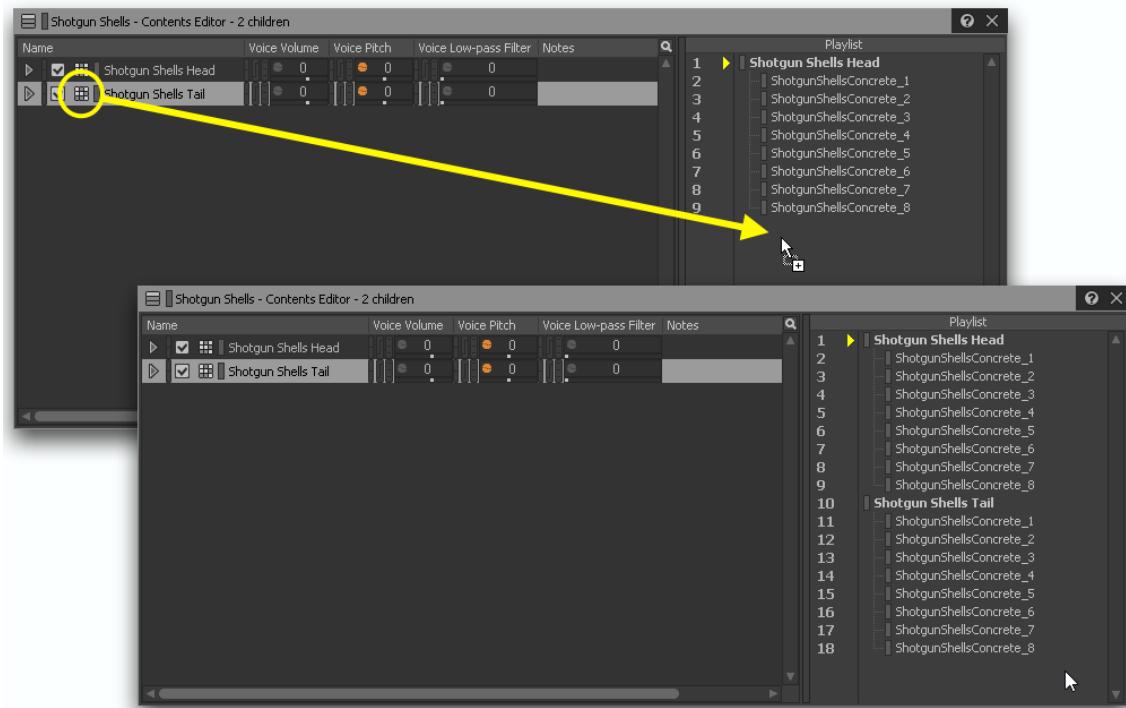
Lesson 2: Designing a Soundscape



Notice that not only do you see the 'Shotgun Shells Head' Random Container, but also you see that the playlist automatically expands to display the Sound SFX objects contained within.

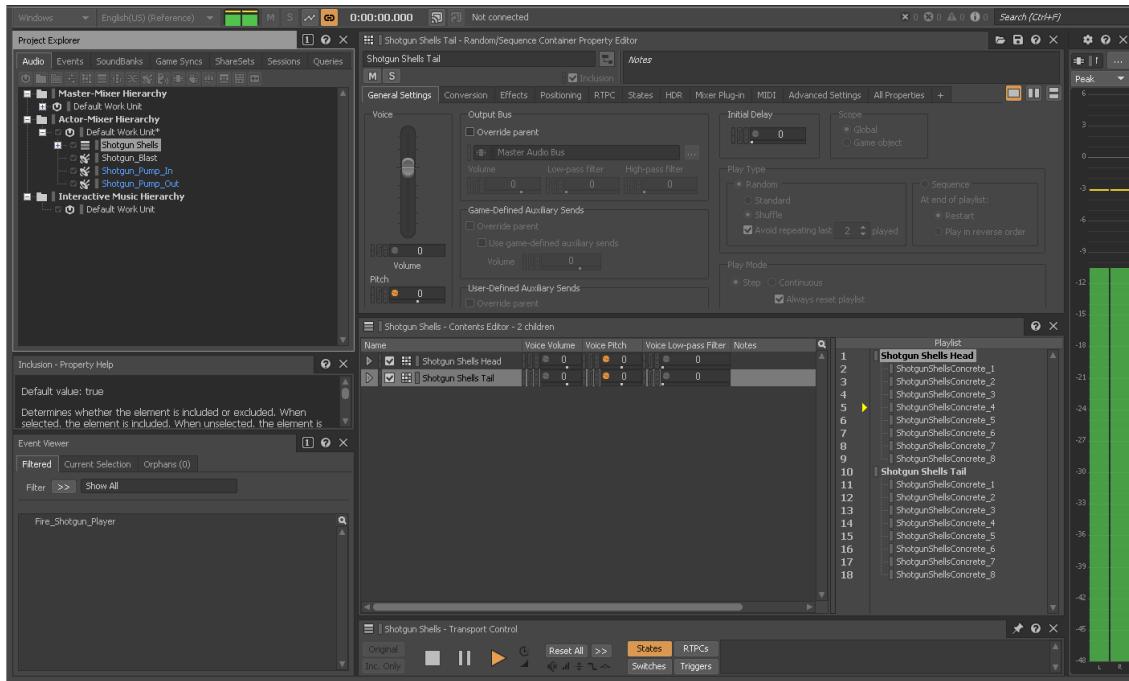
Now you'll add the 'Shotgun Shells Tail' Random Container at the end of the playlist, so that the heads play first, followed by the tail.

6. Drag the Shotgun Shells Tail object to the lower half of the playlist.



Lesson 2: Designing a Soundscape

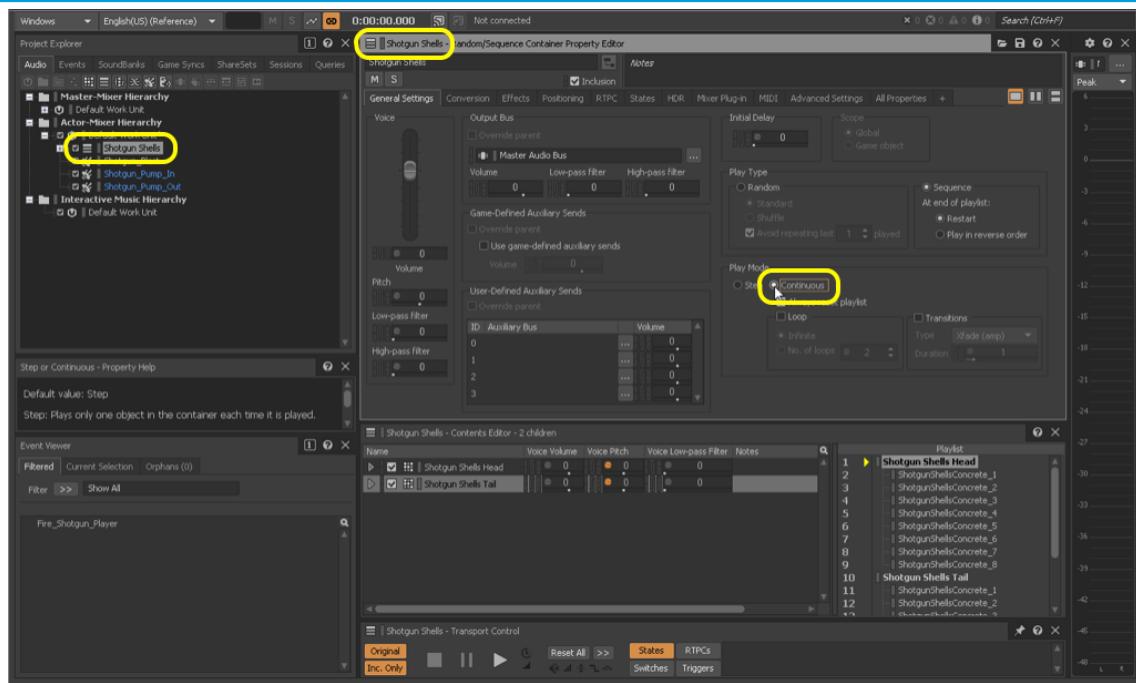
7. Double-click the 'Shotgun Shells' Sequence Container, and then play the object multiple times.



Notice that only one of the head objects plays, and the tail object won't sound until you press play again. If left unchanged, the same thing would happen in the game, that is, the player would fire the shotgun once and only hear the head, and then when they fire a second time they would then hear the tail. This is not the sound sequence you want, but it can be fixed by changing the Play Mode from Step to Continuous.

8. Be sure that the Shotgun Shells Sequence object is selected and click the Continuous radio button.

Lesson 2: Designing a Soundscape



9. Play the Shotgun Shells Sequence Container multiple times.

You now hear a randomly selected head shell followed by a randomly selected tail sound each time you play the object.

Related Video

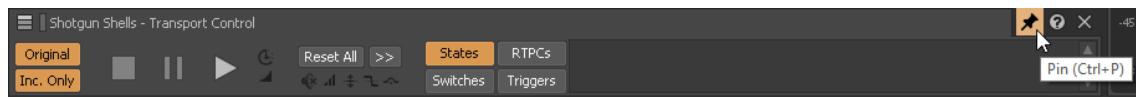
[Wwise-101-04 - Sequencing Sound](#)

Using Silence

The head and tail shell sequence does not sound quite right and this is because there is no space or silence between the initial head sound and the tail sound. Had you left silence either after the head or before the tail sounds when you did your edits earlier, you wouldn't have this problem. However, rather than going back and changing those edits, you'll use a different approach that provides more flexibility.

You're going to be creating and working with another object; however, you're going to need to audition how you use this object within the context of the overall Shotgun Shells Sequence Container. To ensure that the Transport Control stays connected to the Shotgun Shells object, you'll use the Transport Control's Pin feature. This effectively locks the transport to the currently selected object.

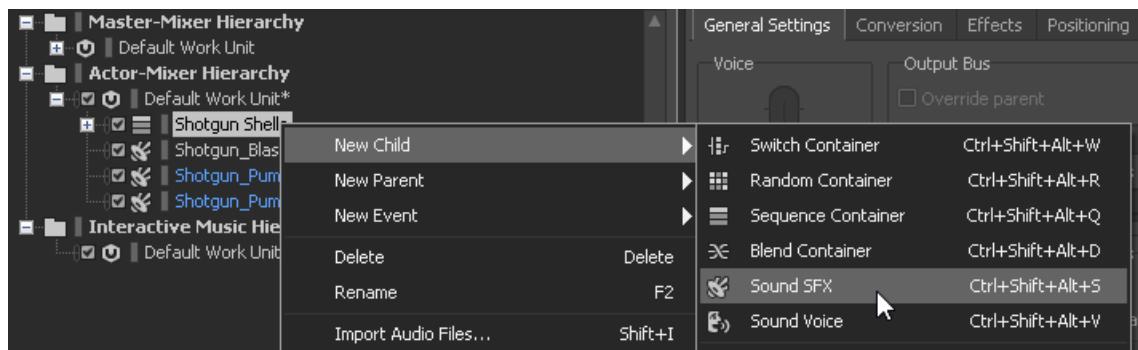
1. Click the Pin in the transport to lock the transport to the Shotgun Shells object.



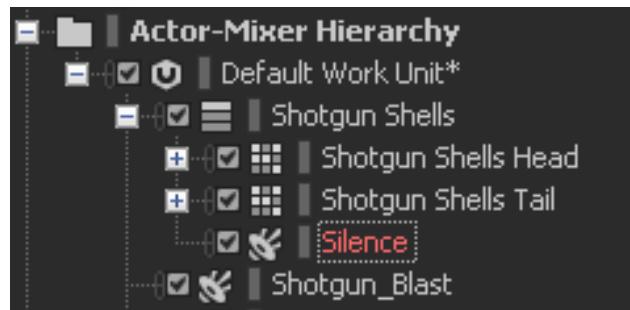
Lesson 2: Designing a Soundscape

To add silence in between the shell head and shell tail, you're going to create an object that will produce only silence and insert it in-between the head and tail objects in the Sequence Container's playlist.

2. Right-click the Shotgun Shells sequence object and choose **New Child > Sound SFX**.



Name your new object Silence. The name is in red because there's no associated source (file or generator) feeding the Sound SFX object.

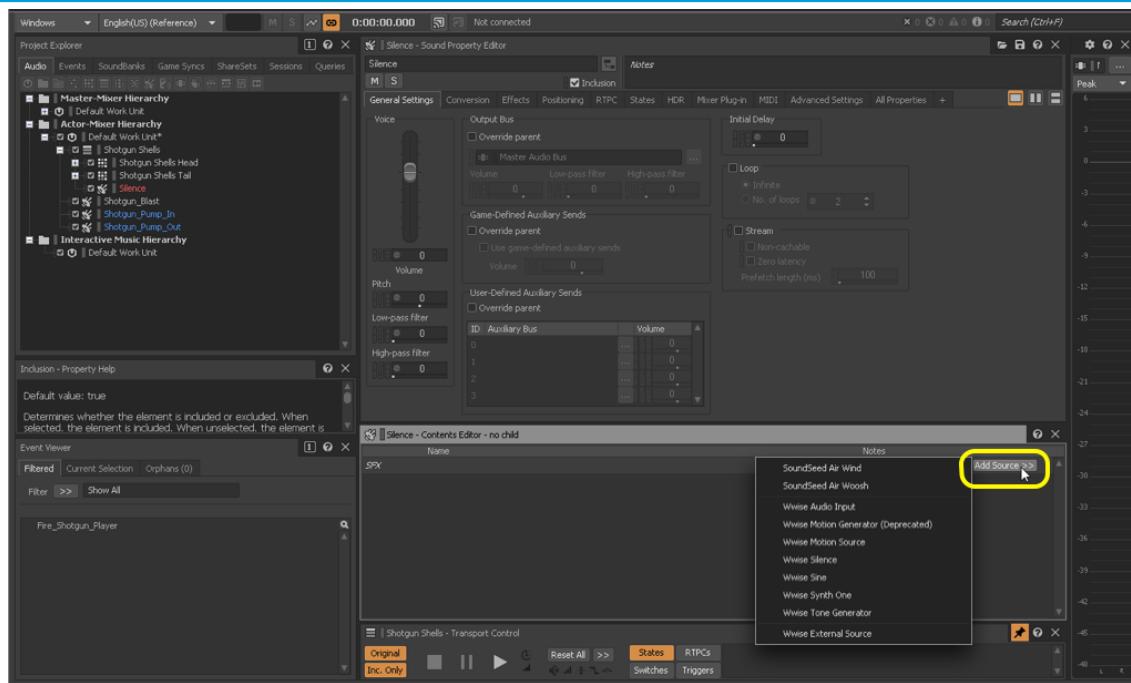


The object's name appears in red, indicating that there's no valid audio source to play through the object.

Up to this point, you've used audio files as the sound source for Sound SFX objects, but there are other types of sources that are available, such as the Synth One synthesizer that generates its own sounds from scratch. In this case, you're going to choose Wwise Silence as the sound (or lack of sound) source.

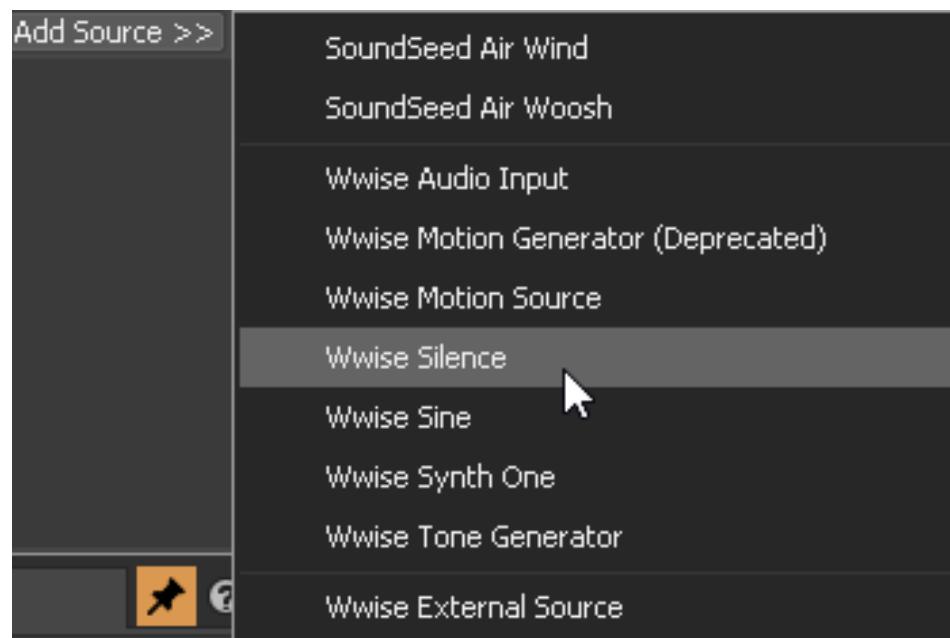
3. In the Silence object's Contents Editor, click **Add Source**.

Lesson 2: Designing a Soundscape



A list of available audio sources is displayed.

4. Choose Wwise Silence.



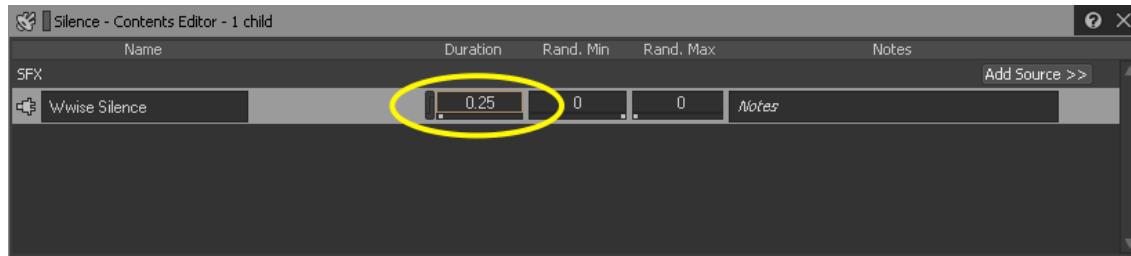
A new Wwise Silence audio source object appears in the Contents Editor and the Silence Sound SFX object in the Project Explorer is no longer red because it now has a valid source.

Lesson 2: Designing a Soundscape



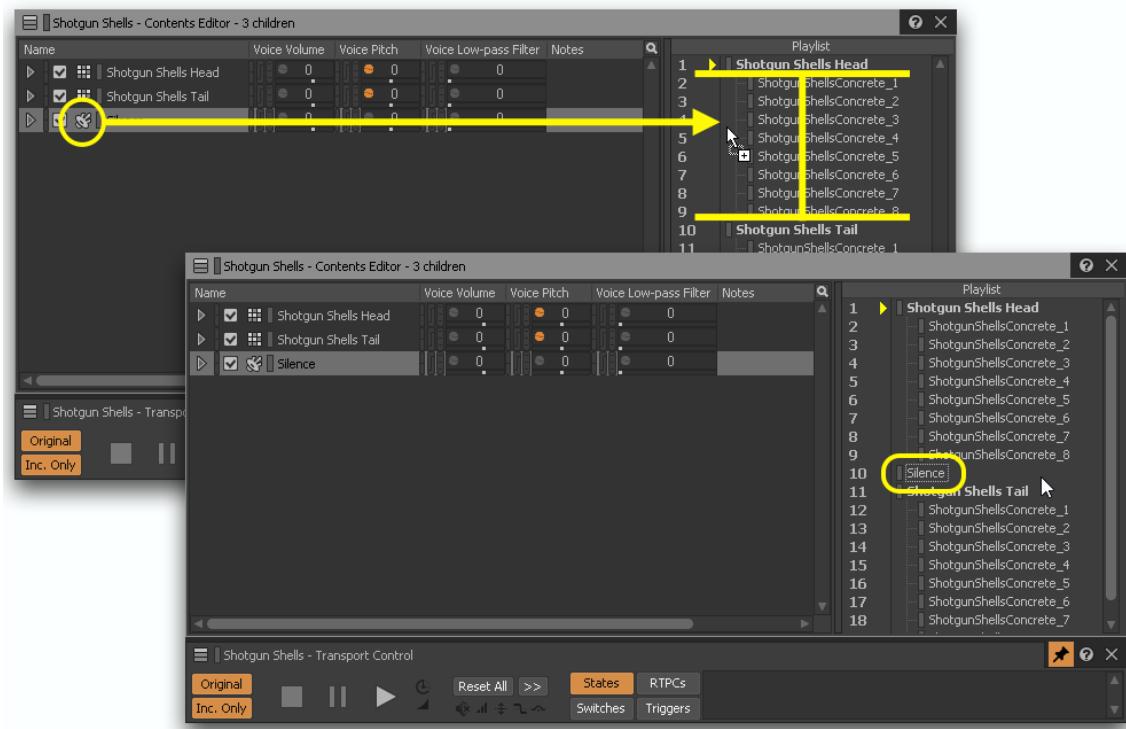
When choosing a new child object you'll see various sound source options, including Wwise Silence listed below the Wwise object types. Selecting one of the sound source options will create the necessary Sound SFX object and assign it to the sound source in a single step.

Wwise Silence provides a parameter for duration. By default it is set to one second. Change the duration to 0.25 seconds.



You now need to add the Silence object to the Sequence Container so that it occurs between the Shotgun Shell Head and Shotgun Shells Tail objects in the playlist. New objects can be added within the playlist of the container objects. So to add silence after the Shotgun Shells Head, you need to drag the Silence object anywhere on lines 2-9.

5. Drag the Silence object anywhere on lines 2-9 in the Playlist.



Lesson 2: Designing a Soundscape

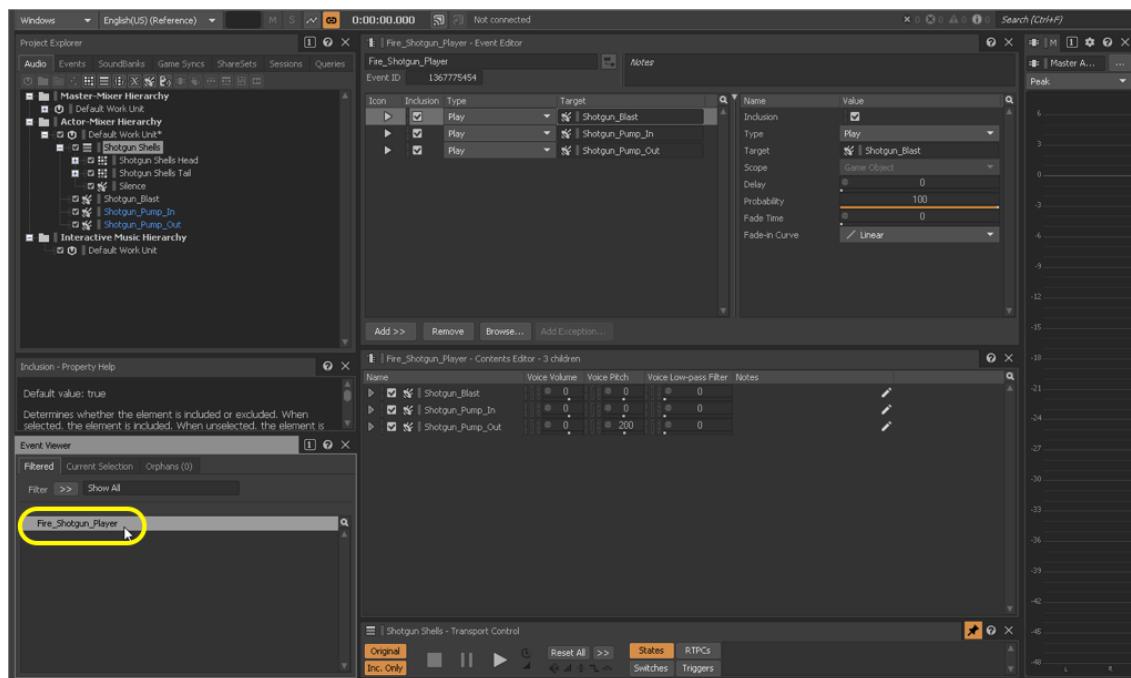
You now see the Silence object situated in between the shell head and tail Random Containers.



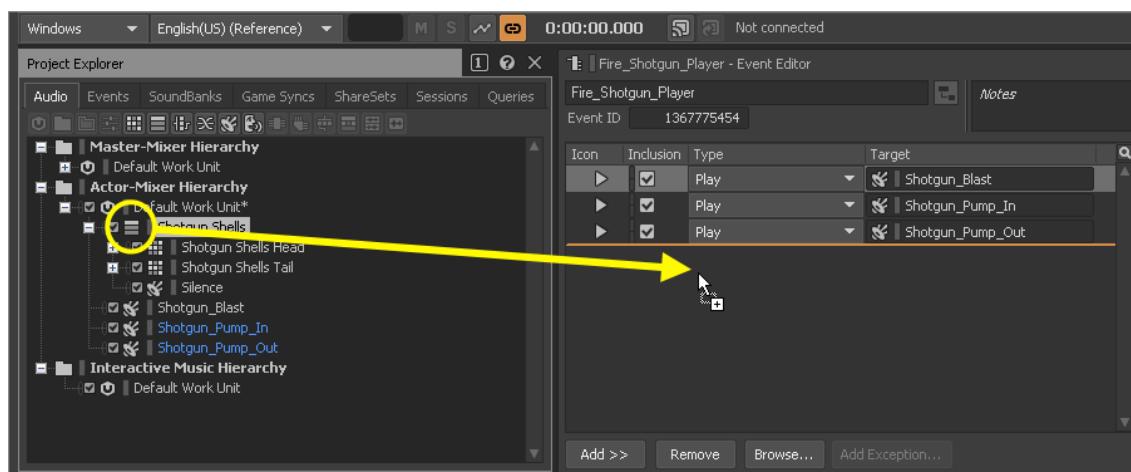
The amount of silence can be adjusted by changing the duration property of the Silence sound source.

Now you need to connect the entire shells sequence to the Fire_Shotgun_Player event in order to hear how it all sounds together.

6. In the Event Viewer, select the Fire_Shotgun_Player event.



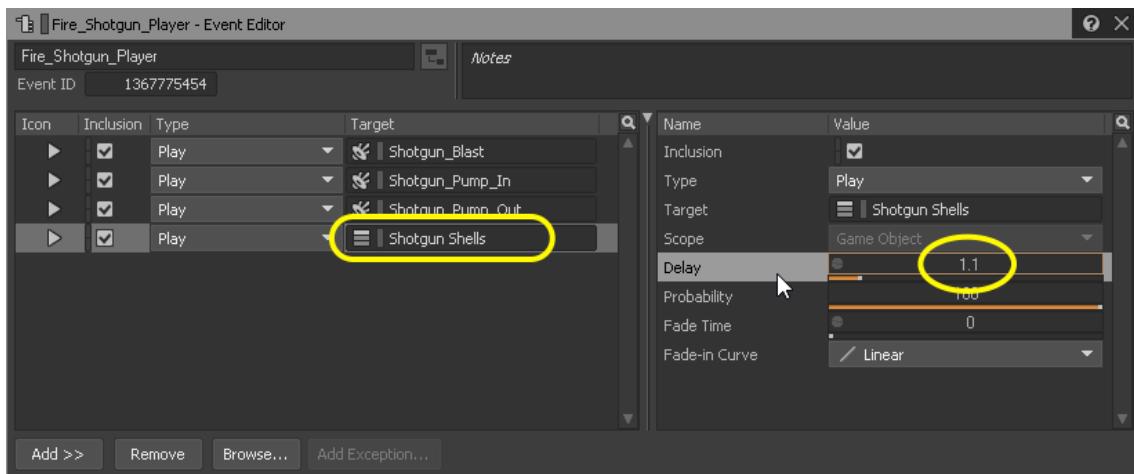
7. Drag the Shotgun Shells Sequence object into the Fire_Shotgun_Player Event Editor.



Lesson 2: Designing a Soundscape

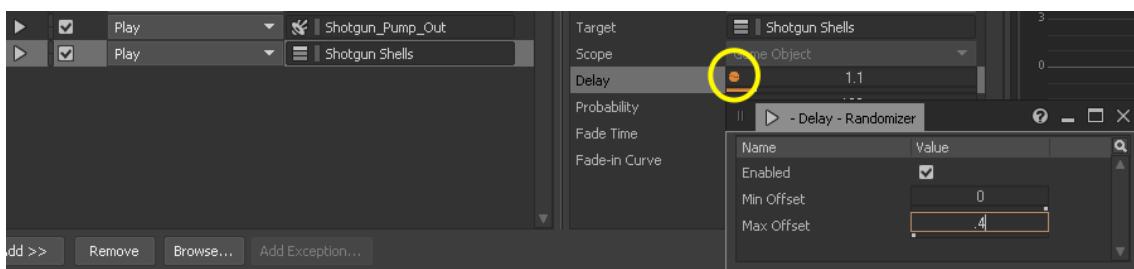
In this configuration, the sound of the shells hitting the ground will be heard before the shotgun pump-in sound is heard, which is not possible with a real shotgun. To avoid this, you'll need to delay the shells.

8. Set the Delay parameter for the newly created Action to 1.1 seconds.



When shells are ejected from a shotgun, they fly in different directions and don't always take the same amount of time before hitting the ground. You can simulate this by randomizing the delay property.

9. Randomize the Delay time by enabling the 'Randomizer' Effect and setting the max value to 0.4 seconds.

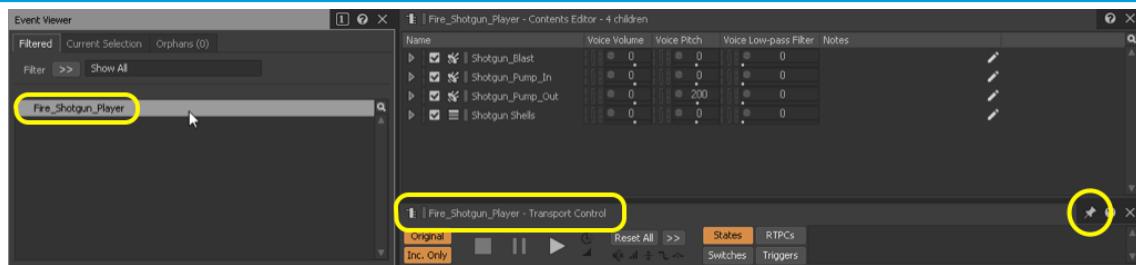


Remember, the values here are offsets to the original value, so in this case it could take up to 1.5 seconds, but no less than 1.1 second before a shell hits the ground. However, since you have no minimum offset, no shell will hit the ground less than .7 seconds from when the trigger is pulled.

Earlier, you pinned the transport to the 'Shotgun Shells' Sequence Container. You'll have to first unpin it in order to focus the transport on the Fire_Shotgun_Player event so that you can hear the entire collection of shotgun related sounds.

10. Unpin the Transport Control and in the Event Viewer, choose the Fire_Shotgun_Player Event and play it to audition the completed Shotgun sequence.

Lesson 2: Designing a Soundscape



Verifying Your Work

1. In the main menu bar choose Views > Capture Log. In the toolbar, click Start Capture and press play on the 'Fire_Shotgun_Player' Event one time, then stop the capture.

Timestamp	Type	Description	Wwise Object Name	Game Object Name
1:27:04.672	Event	Event Triggered (in Wwise)	Fire_Shotgun_...	Transport/Soundcaster
1:27:04.672	Action Triggered	Play	Shotgun_Blast	Transport/Soundcaster
1:27:04.672	Action Delayed	Play, Delay: 400 ms	Shotgun_Pump...	Transport/Soundcaster
1:27:04.672	Notification	Delay Started	Fire_Shotgun_Pl...	Transport/Soundcaster
1:27:04.672	Action Delayed	Play, Delay: 700 ms	Shotgun_Pump...	Transport/Soundcaster
1:27:04.672	Notification	Delay Started	Fire_Shotgun_Pl...	Transport/Soundcaster
1:27:04.672	Action Delayed	Play, Delay: 1290 ms	Shotgun Shells	Transport/Soundcaster
1:27:04.672	Notification	Delay Started	Fire_Shotgun_Pl...	Transport/Soundcaster
1:27:04.693	Notification	Playing	Shotgun_Blast	Transport/Soundcaster
1:27:05.066	Notification	Delay Ended	Fire_Shotgun_Pl...	Transport/Soundcaster
1:27:05.066	Action Triggered	Play	Shotgun_Pump...	Transport/Soundcaster
1:27:05.088	Notification	Playing	Shotgun_Pump...	Transport/Soundcaster
1:27:05.333	Notification	End Reached	Shotgun_Pump...	Transport/Soundcaster
1:27:05.365	Notification	Delay Ended	Fire_Shotgun_Pl...	Transport/Soundcaster
1:27:05.365	Action Triggered	Play	Shotgun_Pump...	Transport/Soundcaster
1:27:05.365	Notification	Playing	Shotgun_Pump...	Transport/Soundcaster
1:27:05.525	Notification	End Reached	Shotgun_Pump...	Transport/Soundcaster
1:27:05.962	Notification	Delay Ended	Fire_Shotgun_Pl...	Transport/Soundcaster
1:27:05.962	Action Triggered	Play	Shotgun Shells	Transport/Soundcaster
1:27:05.962	Notification	Playing	ShotgunShellsC...	Transport/Soundcaster
1:27:06.368	Notification	End Reached And Continue	ShotgunShellsC...	Transport/Soundcaster
1:27:06.378	Notification	Play Continue	ShotgunShellsC...	Transport/Soundcaster
1:27:06.720	Notification	End Reached	Shotgun_Blast	Transport/Soundcaster
1:27:06.730	Notification	End Reached	ShotgunShellsC...	Transport/Soundcaster
1:27:06.730	Notification	Event Finished		Transport/Soundcaster

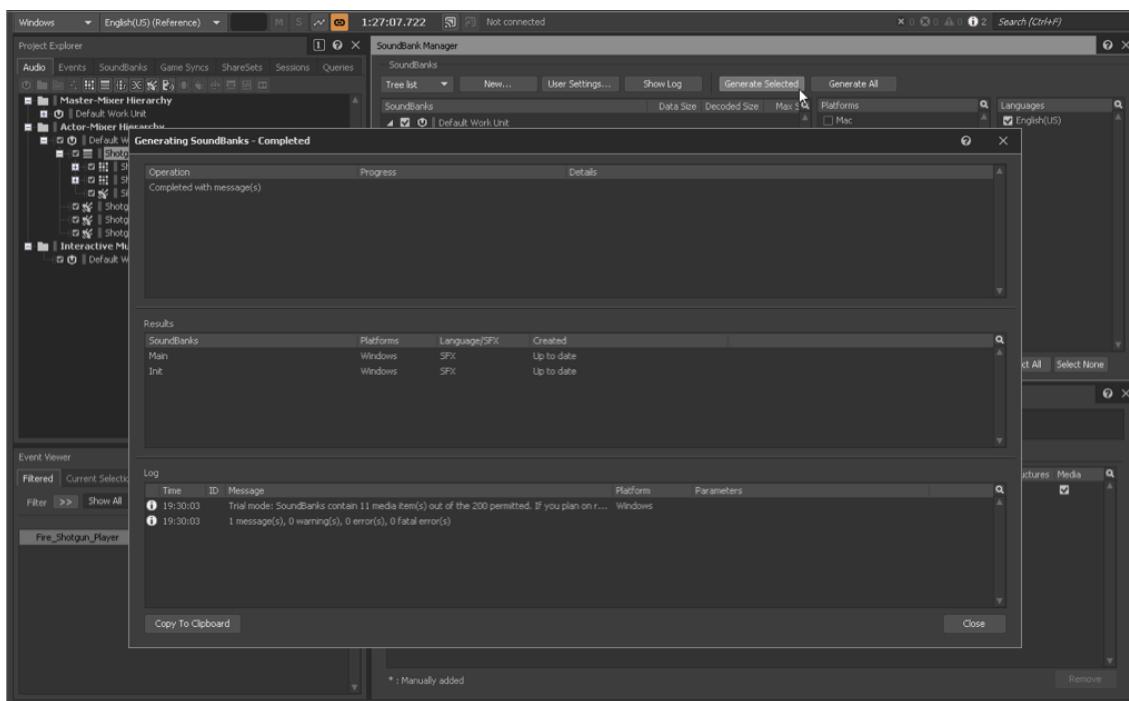
A wealth of information is displayed. At the very top is the initial event game call that's transmitted when you press play, or in game-play when the player pulls the trigger. After that you see various actions that are triggered, which include both play and delay actions. Notifications verify that the intended

Lesson 2: Designing a Soundscape

action is occurring or has finished. This complex series of actions is what produces the soundscape you've crafted in this lesson!

Now you need to build the code based on your sound design and move the sounds into the game itself. This can be done with the push of a button. As you did in the first lesson, you'll go to the SoundBank Layout to generate your SoundBank. This time, instead of selecting the SoundBanks, Platform and Languages to generate, you'll simply use the Generate All feature save some steps.

2. Close the Capture Log, then in the main menu bar choose **Layouts > SoundBank** and click **Generate All**.



3. Launch Cube and play the game and listen to your sound design within game play!

Now you hear how a single event can be used to trigger a lot of sound variation. You can probably even realize that with just this shotgun example alone, you could go back and create even more variety and realism by expanding on the granularization concept you've now put to practice. If all of this can be created from a single event, imagine how quickly a project can grow with hundreds or thousands of events! But there are more message types coming from the game than just events, as you'll discover in the next lesson.