LIMBO

Introduction and Guidelines for LIMBO’s Wwise Project

Contents

Foreword

Welcome to the Limbo game and Wwise project created by the Playdead team and sound designer Martin Stig Andersen. The Limbo game and its corresponding Wwise project have been made available to the Wwise community to allow you to experiment with interactive audio using Wwise along with an actual game. By using the project along with the game, you can monitor in real-time what happens under the hood in an actual game situation. You can also modify the current game sound design by changing audio behaviors, remixing, and replacing the original audio files. You will be dependent, however on the current events and game syncs coded into the game. These are the only restrictions in your new sound design project for the Limbo game.

You will be designing your sound using the triggers from events and game syncs, so make sure to read about them in the section on the Wwise Profiler. It indicates how to connect to the game and get a precise overview of the information the game sends to Wwise as you play.

Installing Wwise and Limbo

To install the latest version of Wwise and Limbo:

1. Go to the Audiokinetic website (audiokinetic.com/en/downloads) and download the Wwise setup (web installer).
2. After the download is complete, double-click the installer.
3. From the Audiokinetic Wwise Setup view, select:
    a. Wwise Authoring Binaries (32 or 64-bit depending on the computer)
    b. Limbo
       Note: Create a separate folder that is not located in the C:\Program Files (x86)\Audiokinetic\Wwise to store the Wwise Limbo project.

Other listed components, like the Cube (Demo) or the Wwise Project Adventure Handbook, for example, can also be installed at this point or at a later time.

About the Game Licence

By default, you have access to the Limbo demo version and you can play, monitor, and modify the audio content of the game up to the first encounter with the giant spider. At this point, you have the option to acquire a full licence to unlock the rest of the game so that you can monitor and modify the whole game content. To have full access to the game with a full license key right away, follow this link: http://LIMBOgame.org/store/.

The Wwise Project

This document provides specific information about the Limbo Wwise project for readers who are already familiar with how to use Wwise. It also gives an overview of the project to help out anyone interested in knowing more about the underpinnings of a video game.

At this point, if you are unfamiliar with Wwise, the best course of action is to go through the Wwise Project Adventure document and companion project to learn about Wwise basics and best practices before you step in the Limbo project.

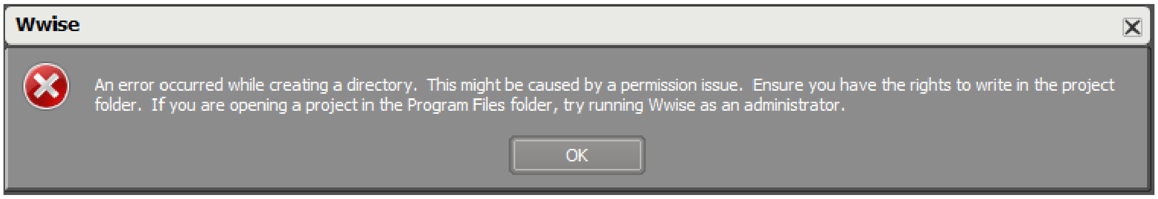
Limbo is a complex project that uses the extensive Wwise feature set to create exceptional sound design and you are free to explore and experiment with the many different approaches that are available to you, both conventional and non-conventional.

This project is provided to you exactly as it was created with Wwise version 2009.3 by the artistic minds of the Playdead's team. Note that some design choices may appear somewhat awkward several years later in light of the newer versions of Wwise.

We hope you'll enjoy exploring this world with all its mystery and ingenuity.

Before You Open the Wwise Project

If you installed Limbo in its default location (under the operating system's Program Files directory) Wwise may not be able to open the Limbo project because of insufficient Windows permissions issues. In this situation, the following error message is displayed when opening the Limbo Wwise project:



An error occurred while creating a directory. This might be caused by a permission issue. Ensure you have the rights to write in the project folder. If you are opening a project in the Program Files folder, try running Wwise as an administrator.

Windows Administrator Error

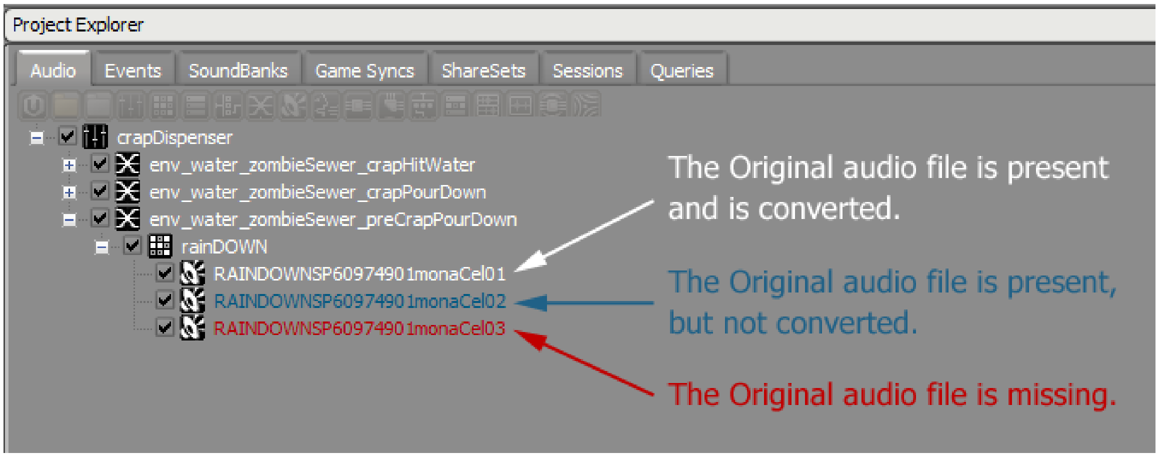
To resolve this issue, we recommend moving the Limbo directory somewhere else on your drive. For example, you can move the "Limbo" folder from "C:\Program Files (x86)\Audiokinetic\Wwise (...)\Limbo" to "C:\Limbo\" and then open the Wwise project from its new location without any problem.

Wwise Project Limitations

To protect Playdead's intellectual property, the included Limbo Wwise project does not contain any of the Original audio files that were used during game creation. This package contains only the converted audio files and game SoundBanks. This means that you can only audition the Limbo "converted" audio files in Wwise.

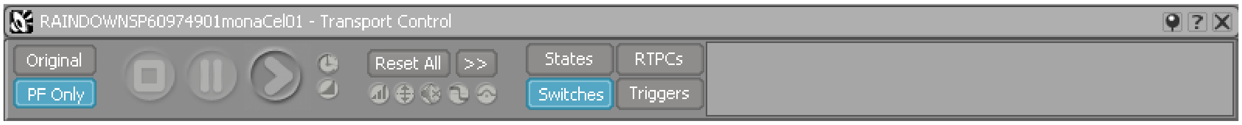
To easily identify if an Original or converted audio file is available in Wwise, the sound object name label has been color coded. If the object name is:

- **White:** The Original audio file is present and was converted, or the sound object contains an audio source plugin.
- **Blue:** The Original audio file is present, but hasn't been converted yet.
- **Red:** The Original audio file is missing, and the sound object may or may not contain a converted audio file.


Project Explorer
Audio
Events
SoundBanks
Game Syncs
ShareSets
Sessions
Queries
crapDispenser
env_water_zombieSewer_crapHitWater
env_water_zombieSewer_crapPourDown
env_water_zombieSewer_preCrapPourDown
rainDOWN
RAINDOWNSP60974901monaCel01
RAINDOWNSP60974901monaCel02
RAINDOWNSP60974901monaCel03
The Original audio file is present and is converted.
The Original audio file is present, but not converted.
The Original audio file is missing.

The color of the object name indicates if the Original or converted audio file is present or not.

By default, the Wwise Transport Control is set up to play the Original audio files. Make sure to turn off the **Original** button on the Transport Control to audition converted audio files instead.



The Original button was turned off to audition the Limbo converted audio files.

> **Note:** If you import other audio files in Wwise, you will have to convert them first if you want to audition them when the **Original** button is turned off. To convert an audio file, you can select **Convert All Audio Files** from the Project menu or right-click the object and select **Convert...** from the Context Menu.

Since the converted audio files and SoundBanks provided in this package are essentials for auditioning the original game audio, we strongly suggest that you backup the files located in the following 2 folders if you ever decide to revert to the original game audio files later on:

- ...*Wwise installation path*\Limbo\PlayGame\data\audio\pc
- ... *Wwise installation path* \Limbo\Resources\audio\Limbo\.cache\Windows\SFX

> **Note:** You can always reinstall the Limbo game using the installer to revert to the Original game audio files and Wwise project, but this will overwrite all your personal modifications to the project if your working copy of the Limbo project was the one located at the default installation location.

Some regular Wwise operations can also delete your existing converted audio files. In a typical game development scenario, these don't affect your project since you already own the Original audio files. Unfortunately when working in the current Limbo project, these operations could also lead to missing converted audio files.

Here's a short list of operations that usually affects converted audio files:

- Routing an object in a HDR bus
- Changing an audio source loop start/end point, fade in/out curve or trimming the audio in the Source Editor view
- Enabling looping in the sound object properties
- Modifying the value in the Conversion Settings
- Enabling Envelope Tracking in the sound object properties
- Enabling Render on an effect in the sound object properties
- Using the Clear Audio File Cache functionality

Abbreviations

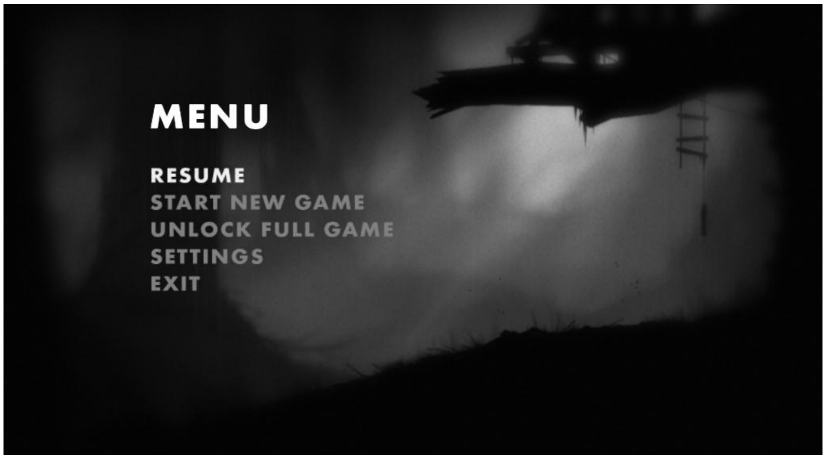
Here is a reference for some abbreviations used in this project that can be useful:

- **RTPC:** Real Time Parameter Control
- **evt:** Event
- **sws:** Switch-State – A Switch conducted by a State Group
- **amb:** Ambience

The Limbo Sound Design

In this section we look at a series of game situations and technical details found in the Limbo game to help you understand how the Limbo sound design was conceived and how the game and Wwise project were built to result in an award winning game.

Menu



Screenshot of the game menu page

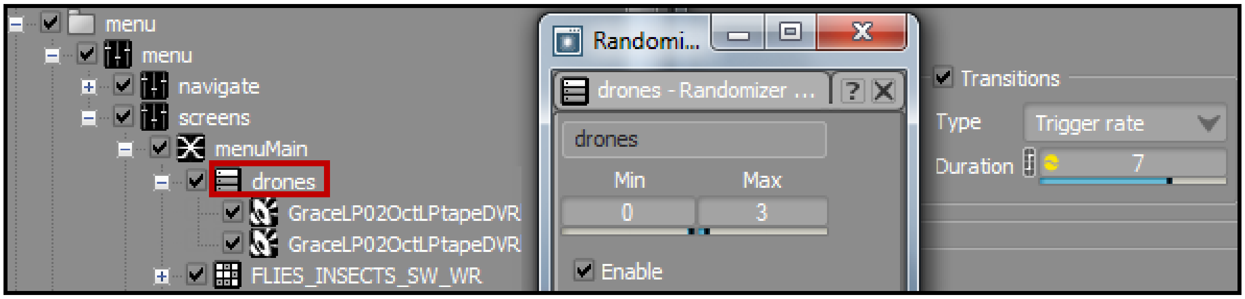
The Limbo game menu uses various techniques to create an audio landscape that never repeats itself. Martin Stig Andersen, the Limbo sound designer, gives a short explanation that sums up the level of details applied to the environmental sound design present in the game:

> *The sounds that you hear on the menu screen are not a loop or anything. These are separate sound files that fade in and out, triggered randomly so that you always get variation.*[1]

Now, let's start by analyzing how the audio for the Limbo menu was created. In the Limbo Wwise project, all menu sounds are included in the 'menu' virtual folder located in the **Audio** tab of the Project Explorer view. The menu is separated into two main components: the background ambience and the navigation sounds.

Drone

The 'drones' sequence container, which is part of the background ambience layer of the menu, is comprised of two sounds meant to overlap each other. To break monotony, the playback time, defined by the 'trigger rate' parameter, has been set to pick a random value between 7 and 10 seconds. Randomizing the start time of each sound in the sequence container ensures the offset between both drones is always different.



drones

Min	Max
0	3

☑ Enable

☑ Transitions

Type: Trigger rate

Duration: 7

Start time is triggered between 7-10 seconds by using a randomizer on the Trigger Rate Duration property of the Sequence Container

Environment

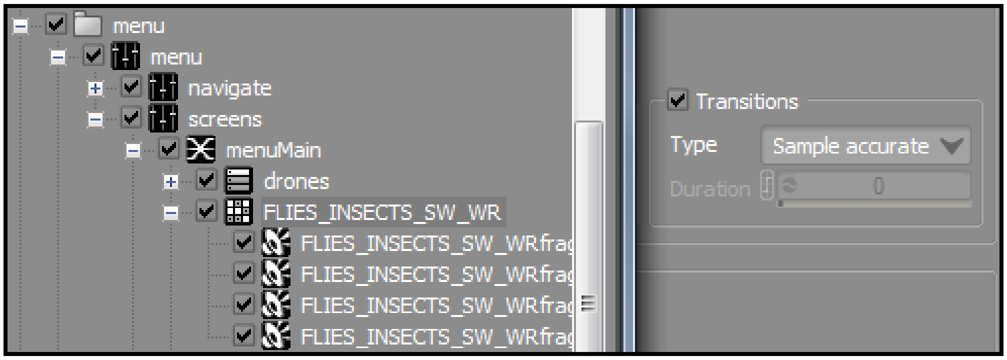
The background and foreground perspective of the environmental sounds of the menu is composed of distinct entities:

- The wind and picture grain (gracetraadnoise) sounds composing the background layer.
- The flies for the foreground layer.

In this section of the project, a sequence container with randomized start times to ensure the resulting audio is always different is used for the wind and the drones,

The picture grain (gracetraadnoise), on the other hand, is using a random container triggering five sound variations sample accurately to ensure that no gap would be present between each fly sample. As you can see in the following image, the flies (FLIES_INSECTS_SW_WR) entity is also using the same mechanism as the picture grain.

[1] *Extract from the article "Q&A: Martin Stig Andersen on Limbo's Soundtrack',* indiegames.com



```
⊟ ☑ menu
  ⊟ ☑ menu
    ⊞ ☑ navigate
    ⊟ ☑ screens
      ⊟ ☑ menuMain
        ⊞ ☑ drones
        ⊟ ☑ FLIES_INSECTS_SW_WR
            ☑ FLIES_INSECTS_SW_WRfrag
            ☑ FLIES_INSECTS_SW_WRfrag
            ☑ FLIES_INSECTS_SW_WRfrag
            ☑ FLIES_INSECTS_SW_WRfrag
```

☑ Transitions

Type: Sample accurate

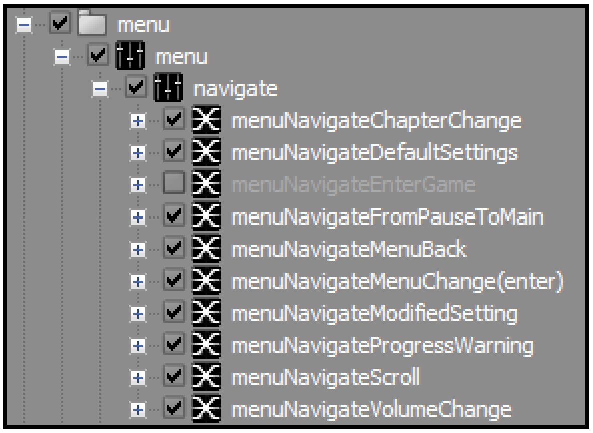
Duration: 0

Fly sounds are triggered by a Random Container using 'sample accurate' transition

Navigation

Navigation sounds provide information to distinguish user interactions like scroll, validation, and back. In the project, the interface sounds are not randomized and always provide the same feedback from the same inputs. In Limbo, blend containers have been put to use to trigger multiple sounds simultaneously.

From an artistic point of view, it's interesting to note that synthesized sounds were used for the navigation to complement the drone layer and to contrast with the more organic aesthetic used for the rest of the game.



```
⊟ ☑ menu
  ⊟ ☑ menu
    ⊟ ☑ navigate
      ⊞ ☑ menuNavigateChapterChange
      ⊞ ☑ menuNavigateDefaultSettings
      ⊞ ☐ menuNavigateEnterGame
      ⊞ ☑ menuNavigateFromPauseToMain
      ⊞ ☑ menuNavigateMenuBack
      ⊞ ☑ menuNavigateMenuChange(enter)
      ⊞ ☑ menuNavigateModifiedSetting
      ⊞ ☑ menuNavigateProgressWarning
      ⊞ ☑ menuNavigateScroll
      ⊞ ☑ menuNavigateVolumeChange
```

Navigation sounds used for menu

In Game

As the menu fades away, you can see a boy lying on his back, slowly waking up, getting ready to start his journey. In this next section, we will go through the various elements of the game that shape the Limbo soundscape.

Footsteps

The first thing you notice in the game is the sound of the boy's footsteps crackling through the windy ambience. As simple as the footsteps sound, they still require a complex mechanism to get to their natural and organic feel. According to Andersen:

> *An idea was to make the world of Limbo silent by making the sound of the boy very loud. In order to keep that from being annoying, I discussed the various ways I attenuated the sound. I*

> *demonstrated how the sounds were triggered in the Limbo editor, the middleware tool created for the sound design, and how those two worked together. These are all attempts to demystify how it's done and show that the sound design and implementation are not really separable.*[2]

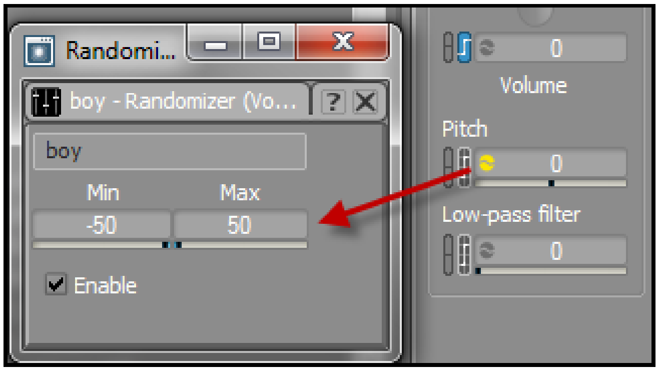
Footsteps are definitely the sounds you hear the most in the game. To increase their variation and reduce redundancy, the following mechanisms have been used:

- The randomization of volume and pitch
- The separation of heel and toe sounds
- The variation of surfaces
- The variation of movement types
- The distance attenuation
- The RTPC curves and states attached to volume

Randomizing Pitch and Volume

Randomizing pitch and volume provides a really efficient way to create diversity with the same set of audio files. Here are two examples applied to the boy sounds (the main character):

We can see a randomizer set on the pitch property , in the "boy" actor-mixer (found in the 'boy' virtual folder located in the Actor-Mixer Hierarchy section of the **Audio** tab of the Project Explorer), that randomize the pitch between -50 to 50 cents each time a children sound from that sound structure is played. Another randomizer has been set on the volume ranging from -6dB to 0dB, on the 'boy_foley_materials' actor-mixer (found under 'boy/boy_foley').



boy

Min -50 Max 50

☑ Enable

Volume 0

Pitch 0

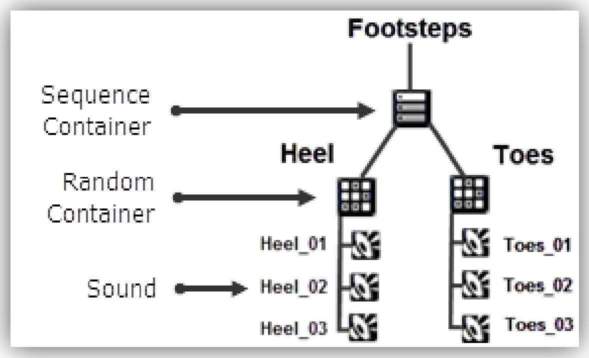
Low-pass filter 0

Randomizer on Pitch

Separating Heel from Toe Sounds

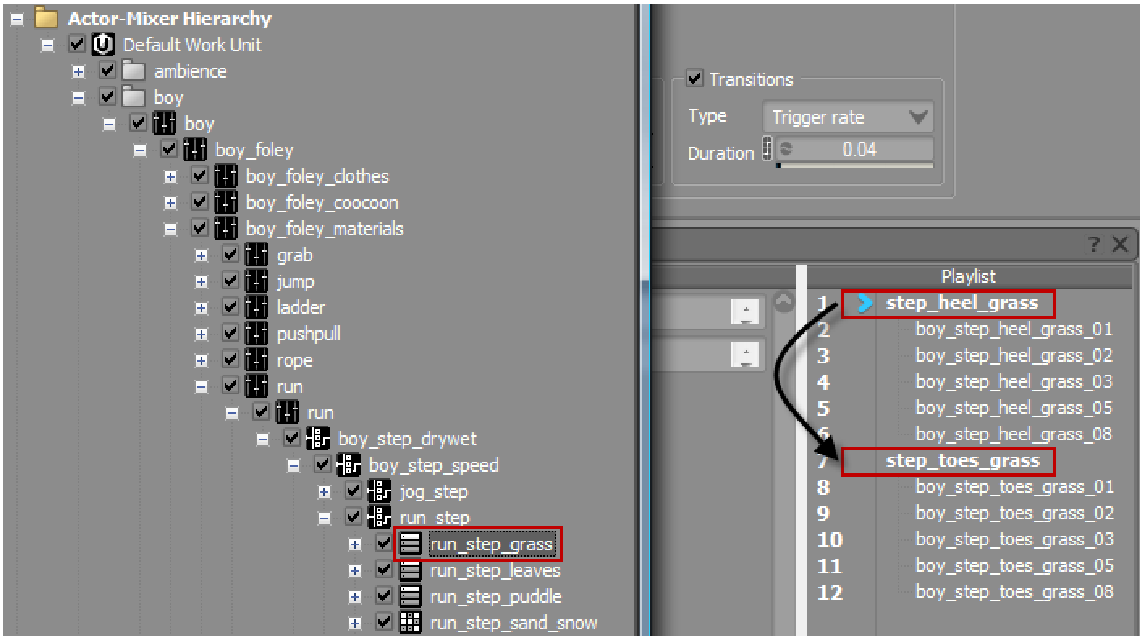
To create diversity heel and toe impact sounds have been separated for each step. To achieve this, the heel and toe sounds have been organized into separate random containers and depending on the surface materials, the random containers are either played from a blend container or played successively from a sequence container.

[2] *Extract from the article "Q&A: Martin Stig Andersen on Limbo's Soundtrack',* indiegames.com



Each footstep splits heel from toe to add diversity

In the following example, the sequence container uses a ‘trigger rate’ transition type set to play the toes random container 0.04 seconds after the heel random container.



☑ Transitions

Type: Trigger rate

Duration: 0.04

Playlist

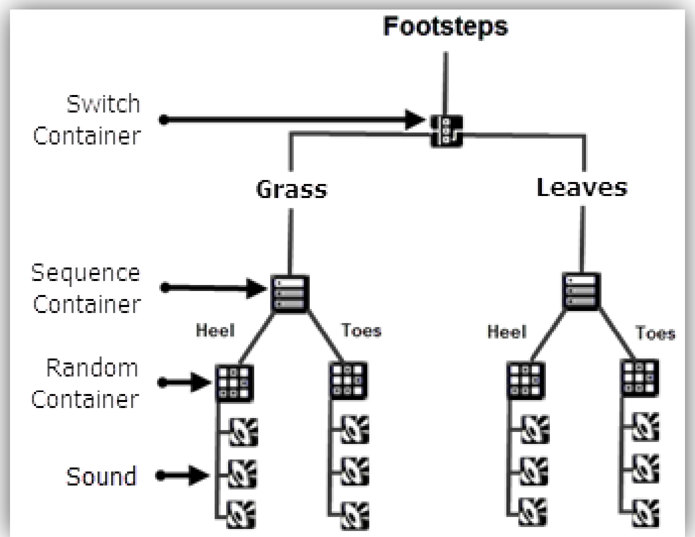
1 step_heel_grass
2 boy_step_heel_grass_01
3 boy_step_heel_grass_02
4 boy_step_heel_grass_03
5 boy_step_heel_grass_05
6 boy_step_heel_grass_08
7 step_toes_grass
8 boy_step_toes_grass_01
9 boy_step_toes_grass_02
10 boy_step_toes_grass_03
11 boy_step_toes_grass_05
12 boy_step_toes_grass_08

Sequence container playlist showing random heels and toes randomization

Variation of Surfaces

The boy's footsteps have been separated by surface types such as grass, leaves, metal, wood, etc. so that different sound ‘textures’ are heard when he walks from one surface type to another. For this purpose, a switch container has been used with a dozen different surface types.

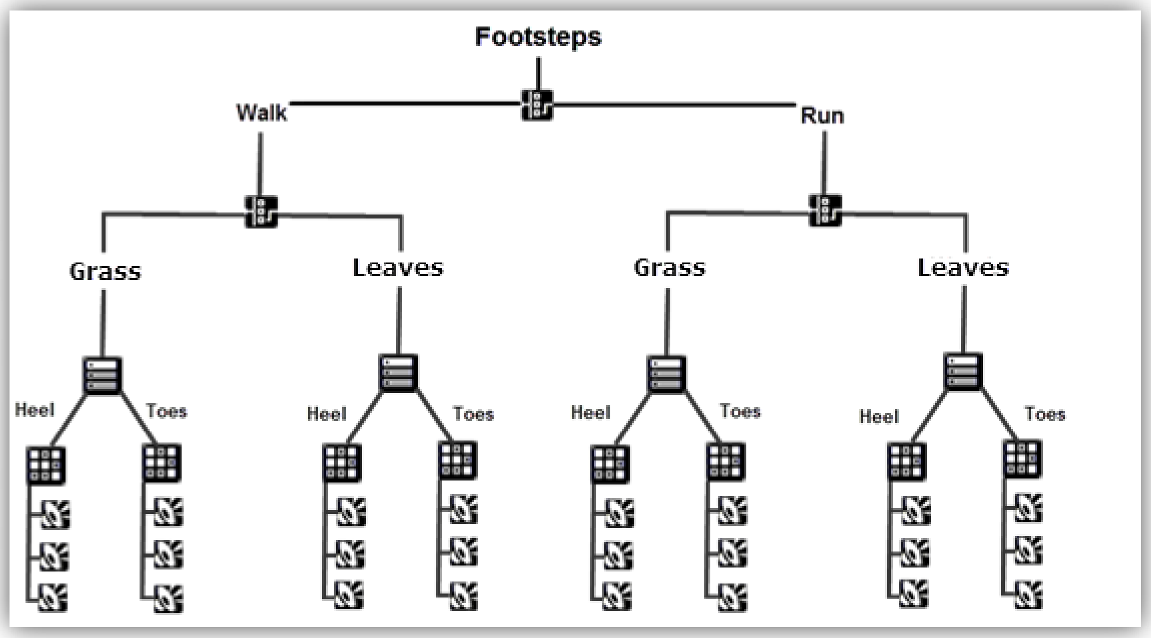
The structure is shown in the following image:



Typical hierarchy used for surface variations in Wwise

Variation of Movement Types

Throughout the game, the boy has four ways of progressing on each surface: by walking, running, jogging, and tiptoeing. Once again, this adds another level of variation to the footstep mechanism that creates even more diversity and helps immerse the player into the story.



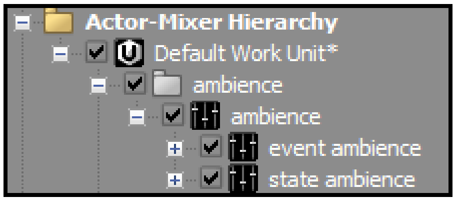
Two levels of switch containers managing surfaces and step types variations

Ambiences

The omnipresence of the background ambience is a key element to Limbo's personality. In the 'ambience' virtual folder, the key elements reside in two separate actor-mixers:

- **Event ambience**: These sounds are triggered by Wwise events and are generally attached to visible game objects like flies or leaves in trees. As a point of reference, the first event ambience triggered by the game is 'amb_start_falling_cliff'.
- **State ambience:** These sounds are room tones used to depict the general atmosphere of the location. They are triggered once at the beginning of the game by the event 'Play_ambience' and then modified over time using States to transition from one location to another. As a point of reference, the first state called by the game is 'amb_sws_start_falling_eyesopen'.

The combination of these two layers creates a constantly changing atmosphere.



- Actor-Mixer Hierarchy
  - Default Work Unit*
    - ambience
      - ambience
        - event ambience
        - state ambience

Overall ambience is composed of event and state ambiences

In the project, event ambiences usually depict visual objects, so they often overlap with other event ambiences. State ambiences, on the other hand cover a broader area of the game and only overlap during the interpolation time when transitioning from one state to another.

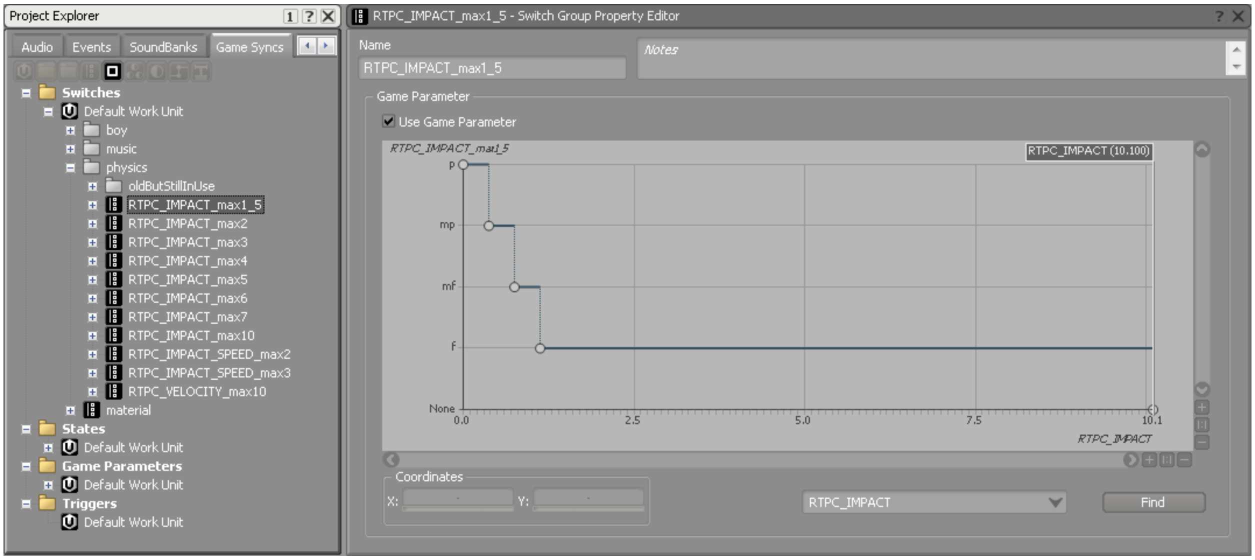
Game Parameters Typical Usage

In Limbo, many game parameters are used to control switch changes and audio properties such as volume, low pass filter, and pitch.

Driving Switch Changes with Game Parameters

For some impact sounds, the game informs Wwise of the velocity of the impact using the 'RTPC_IMPACT' game parameter. For example, the 'RTPC_IMPACT_max1_5' switch group uses the 'RTPC_IMPACT' game parameter to map switches to various velocity ranges:

- **From 0 to 0.375:**-->p 'piano'(light impact)
- **From 0.375 to 0.75:**-->mp 'mezzo piano'
- **From 0.75 to 1.125:**--> mf 'mezzo forte'
- **From 1.125to 10.1:**--> f 'forte' (strong impact)


RTPC_IMPACT_max1_5 - Switch Group Property Editor

Name: RTPC_IMPACT_max1_5

Game Parameter

☑ Use Game Parameter

RTPC_IMPACT

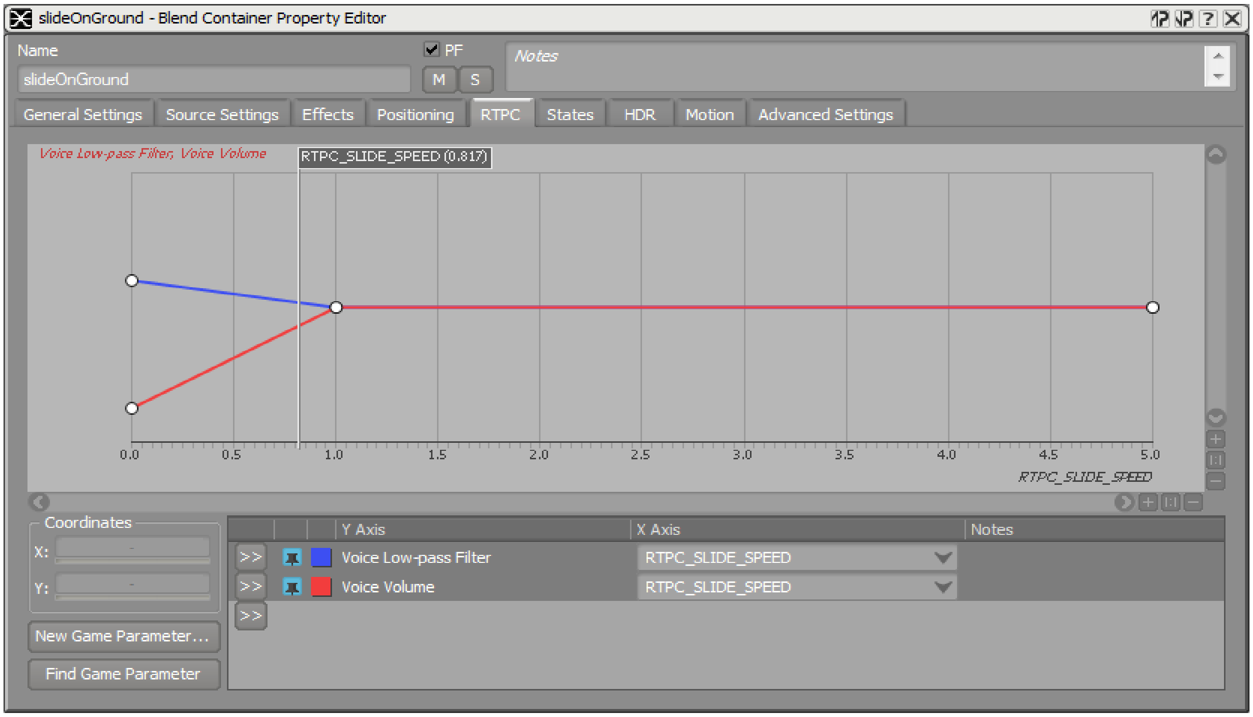
Ranges of the game parameter ‘RTPC_IMPACT’ mapped to various switches

The wagon collision sound, found at the beginning of the game, is a perfect example of a switch driven by a game parameter. The game sends ‘RTPC_IMPACT’ values between 0 and 10.1 which correspond in Wwise to the ‘p’, ‘mp’, ‘mf’ and ‘f’ switches which trigger the expected random containers.

The wagon collision sound in Limbo is using a switch driven by a game parameter

Driving Audio Parameters Using Game Parameters

In Wwise, most audio properties can be controlled by game parameters using RTPC curves. For example, when the boy pulls the boat, the volume and low pass filter parameters of the ‘slideOnGround’ blend container are modified by the curves controlled by the ‘RTPC_SLIDE_SPEED’ game parameter. When the boat stops moving, the velocity of ‘RTPC_SLIDE_SPEED’ game parameter reduces down to zero which modifies the volume and LPF according to the curves built in the RTPC tab of the blend container.



Y Axis	X Axis	Notes
Voice Low-pass Filter	RTPC_SLIDE_SPEED	
Voice Volume	RTPC_SLIDE_SPEED	

Volume and LPF are modified by the ‘RTPC_SLIDE_SPEED’ game parameter when the boy pulls the boat

Be Creative

The easiest way to modify the Limbo sound design is to establish a connection to the game, change the sound properties, and replace the original game sounds with your own audio files.

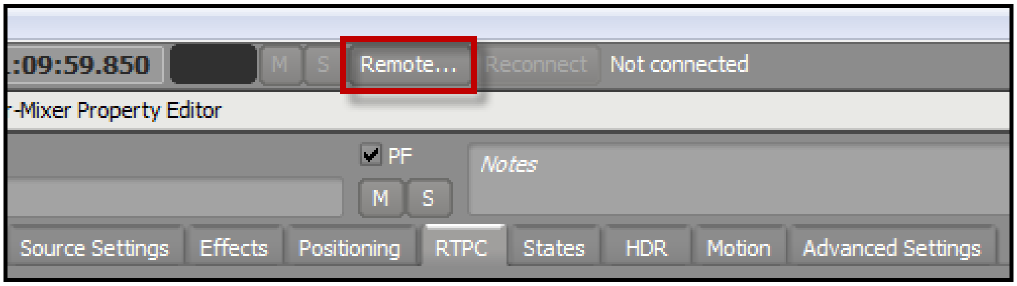
Connecting to the Game

The best way to understand what game elements are communicated to Wwise and how the Limbo sound design was conceived is by connecting the running game to the Wwise authoring tool. Not only can you mix the game audio in real-time, you can also look at the following information sent and received by the game in the Profiler layout:

- The sounds and events triggered and game syncs modifications
- The resource usage such as CPU, memory, streaming, and so on
- The runtime values of properties like volumes, LPF, virtual voices, for each playing voice

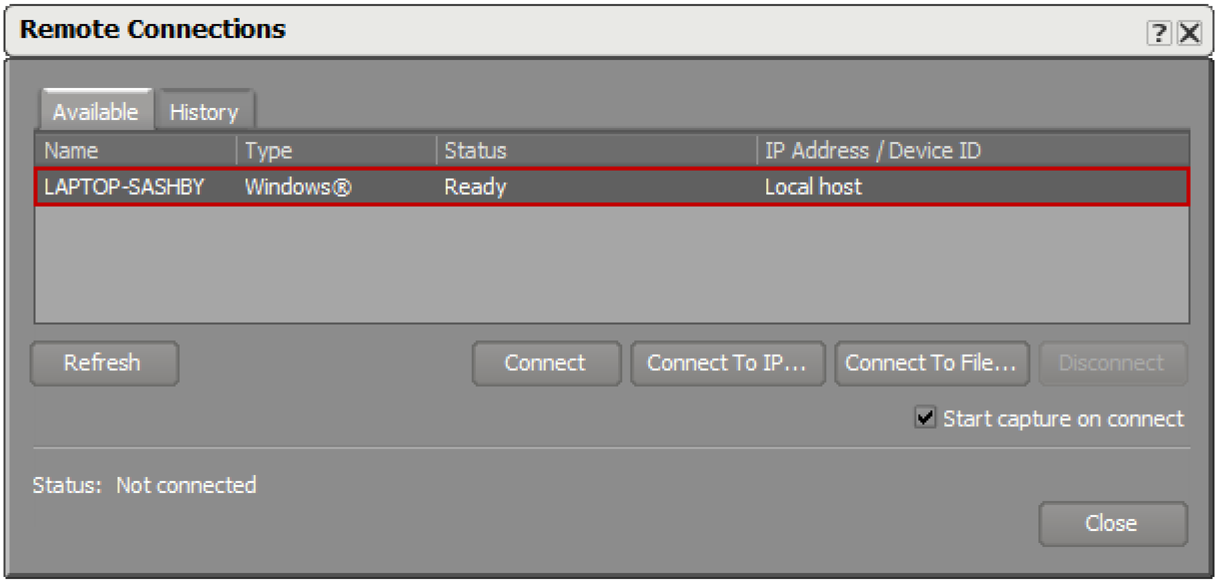
To connect to the game:

1. Make sure that both the Limbo game and Wwise are running.
2. Click **Remote** in the upper right corner of the Wwise authoring application.
   The game running on your computer appears in the Remote Connections view.



Clicking the 'Remote' button on the Wwise toolbar to display the Remote Connections dialog

3. Double-click your computer name to connect to the game or click **Connect**.



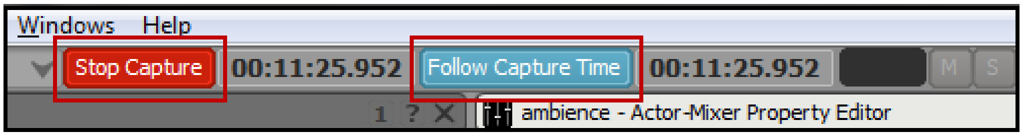
Name	Type	Status	IP Address / Device ID
LAPTOP-SASHBY	Windows®	Ready	Local host

☑ Start capture on connect

Status: Not connected

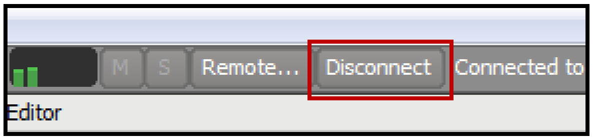
Selecting the computer in the Remote Connections view to connect Wwise to the running Game

4. Once connected, switch to the **Profiler** layout (keyboard shortcut: F6) to monitor what is happening in the game. Information like events, game objects, sounds, and resource usage are monitored in the various views and tabs.
5. Verify that both the **Start Capture** and **Follow Capture Time** buttons are enabled when you profile. Clicking **Capture** starts or stops capturing data from the game and the clicking **Follow Capture Time** updates the data as it's being captured in real time.



The Capture and Follow Capture Time button are enabled to capture the data coming from the connected game

To disconnect Wwise from the Limbo game, click **Disconnect.** You can also quit/exit the Limbo game to automatically disconnect Wwise from the game.



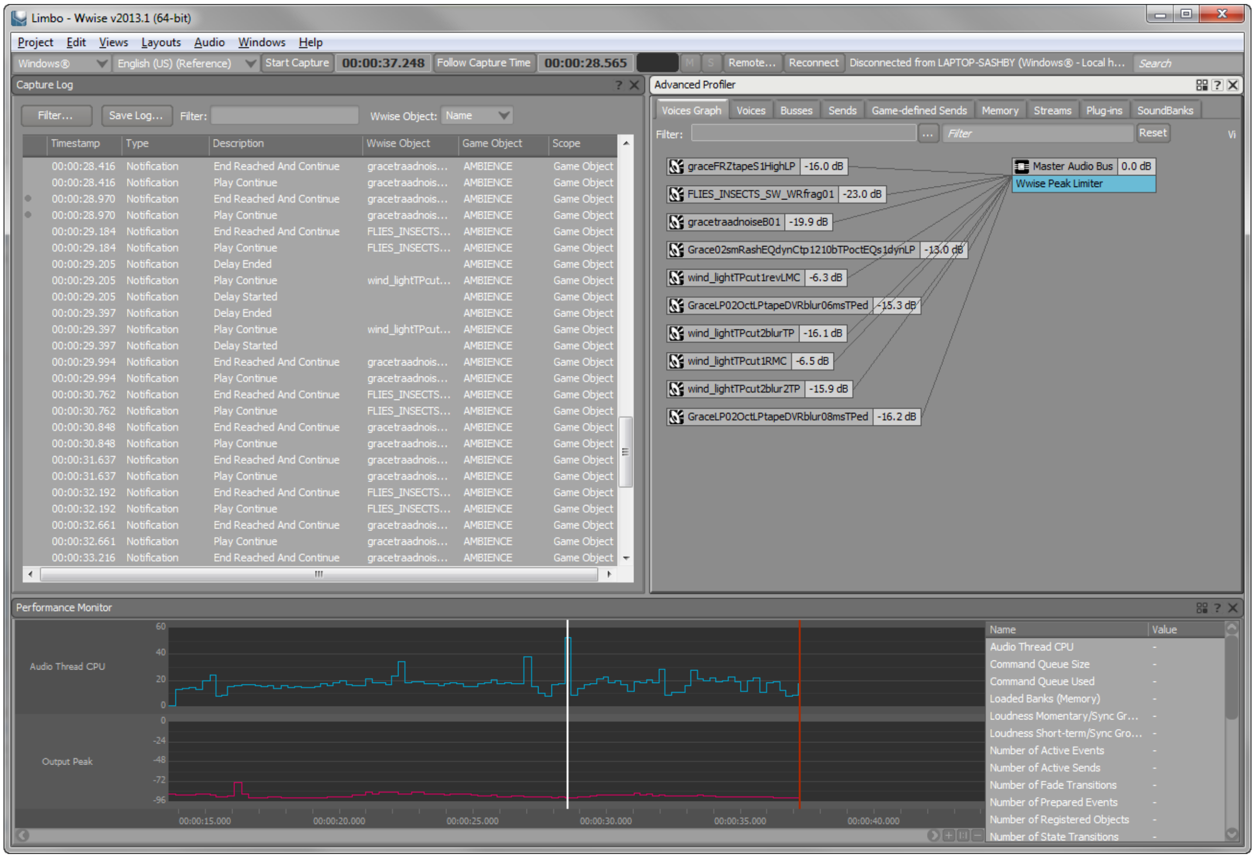
You can press the Disconnect button or close the running game to disconnect it from Wwise

Getting Proficient with the Profiler

Profiler Layout

When opening a project for the first time that was created by someone else, the **Profiler** layout (keyboard shortcut: F6) is your best way to rapidly grasp how the audio was integrated in a game. By default, this layout contains three views: the Capture Log, the Advanced Profiler, and the Performance Monitor.

> **Note:** If you don't see these views, select the **Reset Factory Layout**… option from the **Layout** menu item to bring them back.



Timestamp	Type	Description	Wwise Object	Game Object	Scope
00:00:28.416	Notification	End Reached And Continue	gracetraadnois…	AMBIENCE	Game Object
00:00:28.416	Notification	Play Continue	gracetraadnois…	AMBIENCE	Game Object
00:00:28.970	Notification	End Reached And Continue	gracetraadnois…	AMBIENCE	Game Object
00:00:28.970	Notification	Play Continue	gracetraadnois…	AMBIENCE	Game Object
00:00:29.184	Notification	End Reached And Continue	FLIES_INSECTS…	AMBIENCE	Game Object
00:00:29.184	Notification	Play Continue	FLIES_INSECTS…	AMBIENCE	Game Object
00:00:29.205	Notification	Delay Ended		AMBIENCE	Game Object
00:00:29.205	Notification	Play Continue	wind_lightTPcut…	AMBIENCE	Game Object
00:00:29.205	Notification	Delay Started		AMBIENCE	Game Object
00:00:29.397	Notification	Delay Ended		AMBIENCE	Game Object
00:00:29.397	Notification	Play Continue	wind_lightTPcut…	AMBIENCE	Game Object
00:00:29.397	Notification	Delay Started		AMBIENCE	Game Object
00:00:29.994	Notification	End Reached And Continue	gracetraadnois…	AMBIENCE	Game Object
00:00:29.994	Notification	Play Continue	gracetraadnois…	AMBIENCE	Game Object
00:00:30.762	Notification	End Reached And Continue	FLIES_INSECTS…	AMBIENCE	Game Object
00:00:30.762	Notification	Play Continue	FLIES_INSECTS…	AMBIENCE	Game Object
00:00:30.848	Notification	End Reached And Continue	gracetraadnois…	AMBIENCE	Game Object
00:00:30.848	Notification	Play Continue	gracetraadnois…	AMBIENCE	Game Object
00:00:31.637	Notification	End Reached And Continue	gracetraadnois…	AMBIENCE	Game Object
00:00:31.637	Notification	Play Continue	gracetraadnois…	AMBIENCE	Game Object
00:00:32.192	Notification	End Reached And Continue	FLIES_INSECTS…	AMBIENCE	Game Object
00:00:32.192	Notification	Play Continue	FLIES_INSECTS…	AMBIENCE	Game Object
00:00:32.661	Notification	End Reached And Continue	gracetraadnois…	AMBIENCE	Game Object
00:00:32.661	Notification	Play Continue	gracetraadnois…	AMBIENCE	Game Object
00:00:33.216	Notification	End Reached And Continue	gracetraadnois…	AMBIENCE	Game Object

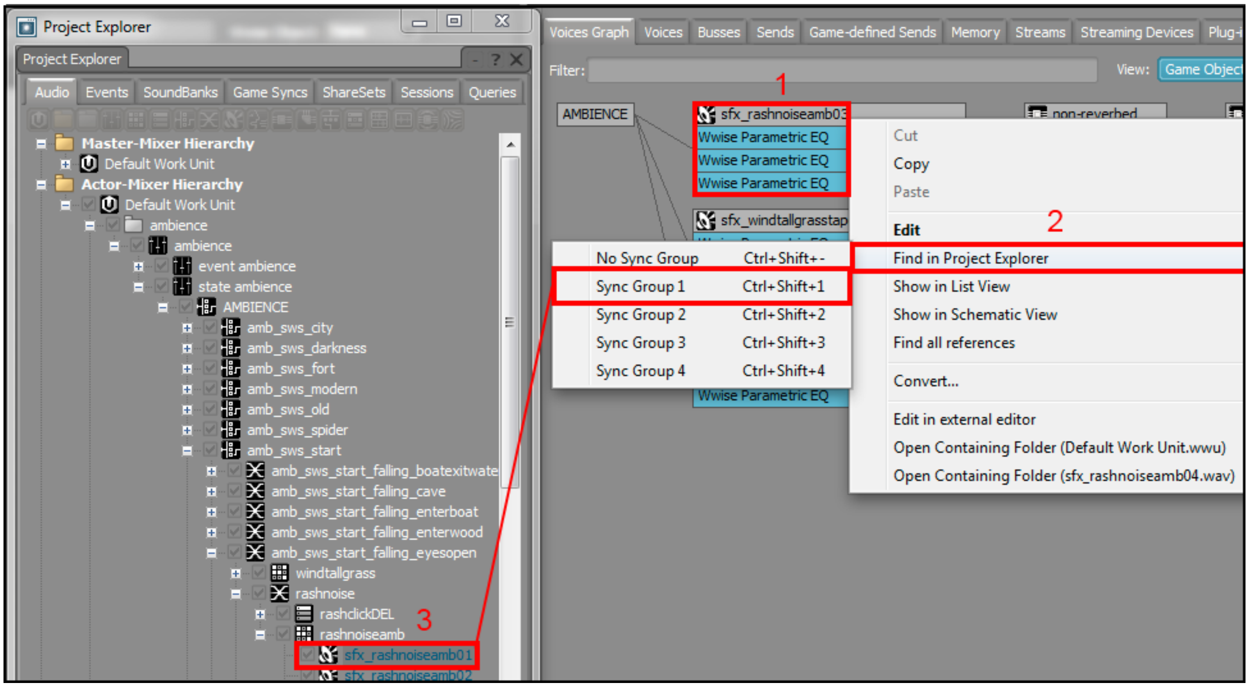
Name	Value
Audio Thread CPU	-
Command Queue Size	-
Command Queue Used	-
Loaded Banks (Memory)	-
Loudness Momentary/Sync Gr…	-
Loudness Short-term/Sync Gro…	-
Number of Active Events	-
Number of Active Sends	-
Number of Fade Transitions	-
Number of Prepared Events	-
Number of Registered Objects	-
Number of State Transitions	-

Some information captured from the game in the Profiler layout

For more information about how to use the Profiler, refer to the Wwise documentation.

Retrieving Objects Displayed in the Profiler

To locate an object in the Project Explorer from the Capture Log, the Advanced Profiler, or any other view, right-click the object and select the **Find in Project Explorer** option. The selected object will automatically be highlighted in the Project Explorer. This is particularly convenient whenever you need to locate an object in the project hierarchy and change its properties.



Right-click and select "Find in Project Explorer" to locate an object in the project hierarchy

To edit any object property, double-click the object to display its properties in the Property Editor view.

You can also easily locate an object by using the **Search** field on the Wwise toolbar and then inspect its properties by selecting it in the results list.

Replacing the Original Sounds

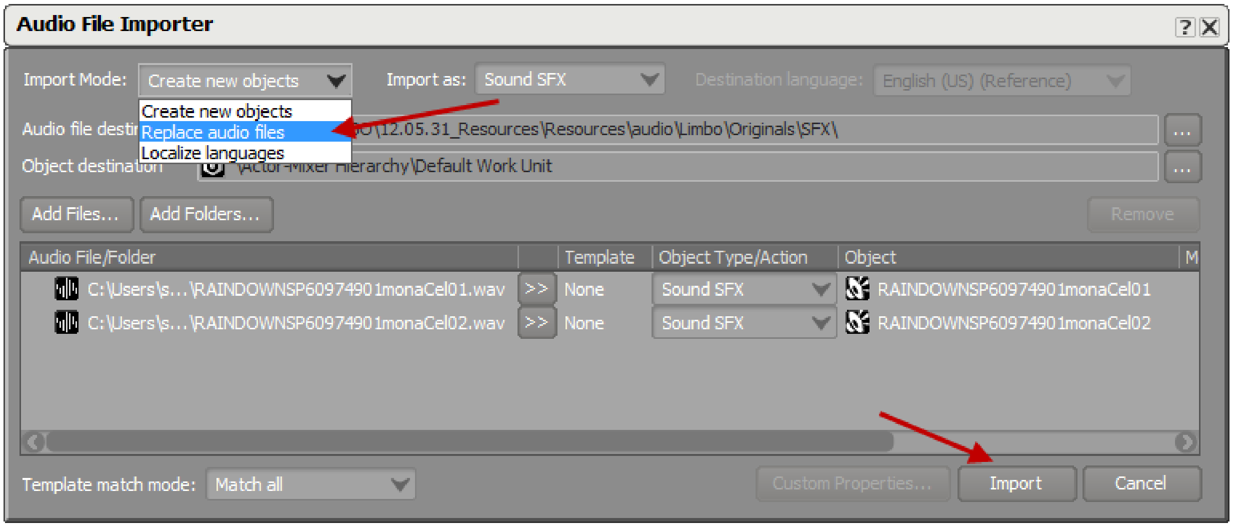
You can replace the default Limbo sounds found in the project and even add new ones from your own collection of audio files. Setting up an alternative sound design for the original Limbo game is an appealing and challenging project for anyone interested in refining their craft and learning more about audio for games.

Replacing Existing Sounds

You can replace existing sounds in two ways depending on how you want to name your files. The methods differ based on whether you want to replace the original audio files by using the same names for your new assets or you want to add your audio files using different names and then assign them as audio sources to sound object.

Using Same Audio File Names

This is the easiest approach. It consists of a simple drag and drop of the new audio files over the top node 'Actor Mixer Hierarchy' in the Project Explorer which will automatically display the Audio File Importer view. In the **Import Mode** drop down menu located in the upper-left corner of the view, select the **Replace audio files** option and click **Import** to replace the existing audio files.



Audio File Importer

Import Mode: Create new objects

Import as: Sound SFX

Destination language: English (US) (Reference)

Audio file desti... ...\12.05.31_Resources\Resources\audio\Limbo\Originals\SFX\

Object destination ...\Actor-Mixer Hierarchy\Default Work Unit

Audio File/Folder		Template	Object Type/Action	Object
C:\Users\s...\RAINDOWNSP60974901monaCel01.wav	>>	None	Sound SFX	RAINDOWNSP60974901monaCel01
C:\Users\s...\RAINDOWNSP60974901monaCel02.wav	>>	None	Sound SFX	RAINDOWNSP60974901monaCel02

Template match mode: Match all

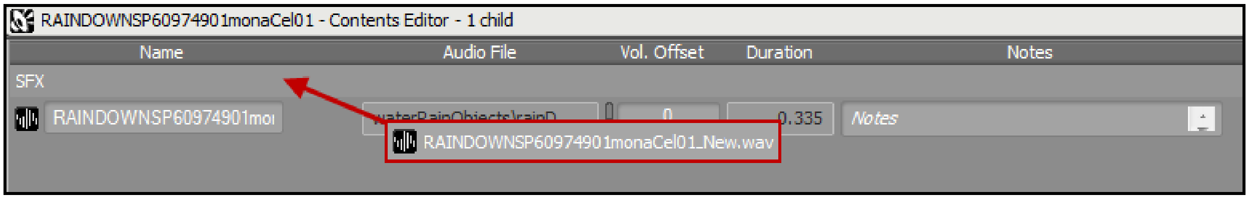
Select ‘Replace audio files’ in the Audio File Importer

Using Different Audio File Names

Using different names for the new audio files does require a few extra steps but provides the advantage of preserving the original sounds from the game which gives you the option of performing A/B comparisons. Comparing the originals and new audio files can be useful in various tasks such as volume matching, comparing durations, and validating aesthetic for example.

To replace an audio file used by a sound with a new one named differently:

1. Make sure that the new audio file name is unique. Adding a postfix to the original audio file name is a good idea. For example, “…_New” or “…_[YOUR INITIALS]”.
2. Locate the sound for which you want to replace the audio file and display its audio source in the Content Editor view (you can use the ‘Designer’ layout for this (keyboard shortcut: F5).
3. Drag and drop the new audio file on the **object type** label located above the audio source you want to replace as shown in the following image.



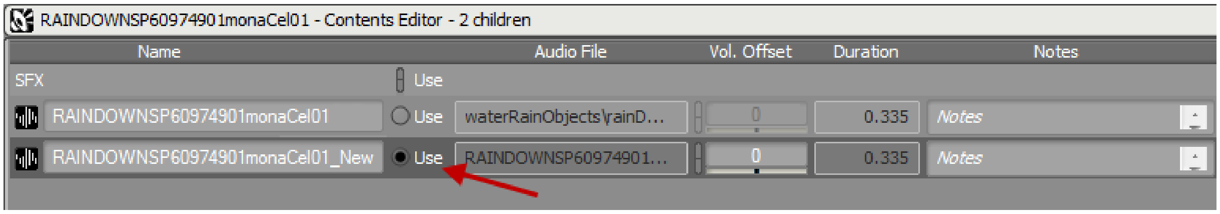
RAINDOWNSP60974901monaCel01 - Contents Editor - 1 child

Name	Audio File	Vol. Offset	Duration	Notes
SFX				
RAINDOWNSP60974901moı	waterRainObjects\rainD...	0	0.335	Notes

RAINDOWNSP60974901monaCel01_New.wav

Drag & drop audio file to 'SFX' bar above the audio source

4. Now that the newly created audio source appears under the original one, select the **Use** radio button on this new audio source. The sound object will now use and play this new audio file from now on.



RAINDOWNSP60974901monaCel01 - Contents Editor - 2 children

Name		Audio File	Vol. Offset	Duration	Notes
SFX	Use				
RAINDOWNSP60974901monaCel01	○ Use	waterRainObjects\rainD...	0	0.335	Notes
RAINDOWNSP60974901monaCel01_New	● Use	RAINDOWNSP60974901...	0	0.335	Notes

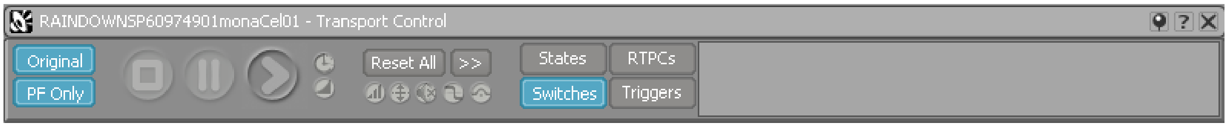
Select 'Use' on the new audio source

Auditioning Your New Sounds in Wwise.

As we mentioned at the beginning of this document, to protect Playdead's intellectual property, the included Limbo Wwise project does not contain any of the Original audio files, but only the converted ones. As a result, you can only audition the Limbo "converted" audio files in Wwise.

Therefore, if you want to audition your newly imported audio files, you have 2 options:

- You can click **Original** in the Transport to audition the Original audio file you just imported. This way you will be able to audition your new audio file right away. To audition the converted audio files already provided in the Limbo project, you need to click **Original** again to switch back to the Limbo project files.



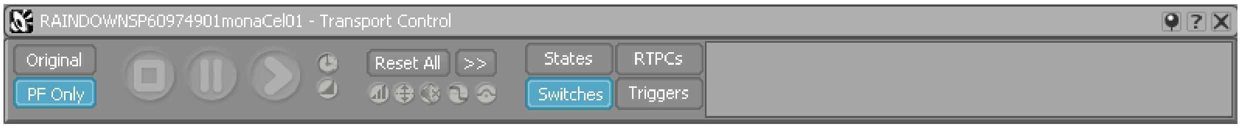
The Original button was turned on to audition the newly imported Original audio file.

- You can convert the audio file you just imported without specifying Original in the Transport to audition your newly converted audio file instead.

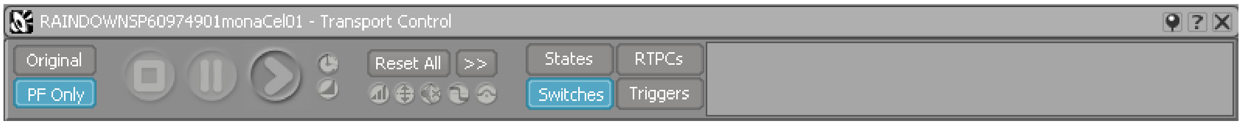
Note: To convert an audio file, you can select **Convert All Audio Files** from the Project menu or right-click the object and select **Convert...** from the Context Menu.

Converting your audio files allows you to audition all the converted audio files provided in the project. This will be particularly useful if you want to perform an A/B comparison between the original sound from the game and your newly imported sound, by toggling the **Use** radio button in the Contents Editor, without having to turn on/off the **Original** button between each playback.

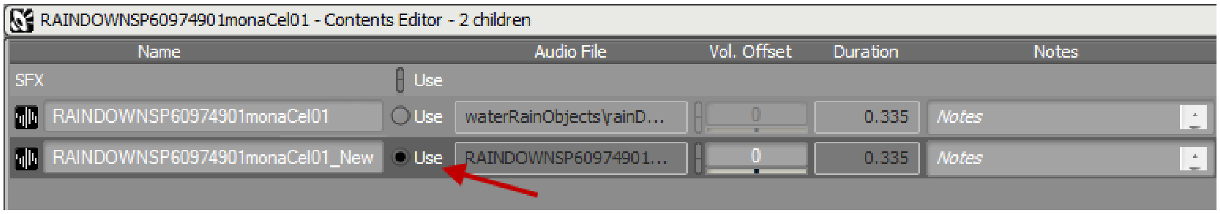
Refer to the Wwise Project Limitations section for more information about auditioning Original or Converted audio files in Wwise.



The Original button is turned off to audition the converted audio file, but the Play button is disabled since the newly imported audio file hasn't been converted yet.



Following the audio file conversion, the Play button becomes enabled to let you audition your newly converted audio file.


RAINDOWNSP60974901monaCel01 - Contents Editor - 2 children

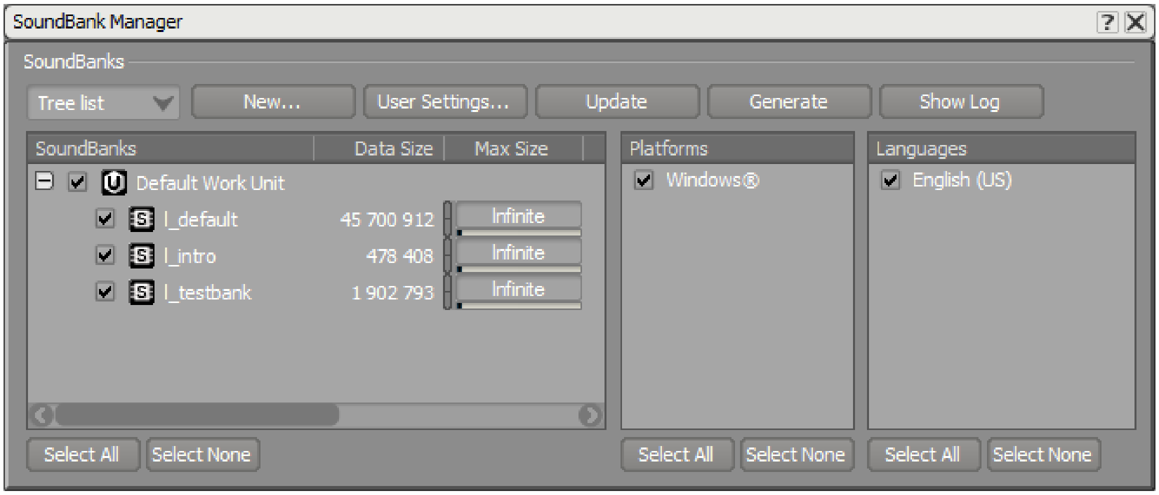
Name		Audio File	Vol. Offset	Duration	Notes
SFX	Use				
RAINDOWNSP60974901monaCel01	○ Use	waterRainObjects\rainD…	0	0.335	Notes
RAINDOWNSP60974901monaCel01_New	● Use	RAINDOWNSP60974901…	0	0.335	Notes

Toggle the Use radio button between playbacks to perform an A/B comparison between the original sound from the game and your newly converted audio file.

Regenerating the SoundBanks

Note: Before you update or regenerate the SoundBanks, it is strongly recommended that you backup the project "Generated SoundBanks" folder if you want the option to revert to the default sounds of the Limbo game eventually.

To hear the new audio files you imported in Wwise in your next Limbo play through, you need to regenerate the SoundBanks that are used by the game. First display the SoundBank Manager view (keyboard shortcut: Shift+B), or switch to the SoundBank layout (keyboard shortcut: F7). Then, make sure to select all the SoundBanks, Platforms and Languages, for which you want to regenerate the SoundBanks and click the **Generate**.



SoundBanks	Data Size	Max Size
☑ Default Work Unit		
☑ l_default	45 700 912	Infinite
☑ l_intro	478 408	Infinite
☑ l_testbank	1 902 793	Infinite

Platforms
☑ Windows®

Languages
☑ English (US)

Select SoundBanks, Platforms and Languages, then use Generate button

Note: When you regenerate the SoundBanks, every property change (for example, volume, RTPC curves) is also packaged along with the new audio files.

Conclusion

Congratulations! You should now be ready to explore the Limbo game on your own and modify the content of the original to your liking. We hope this basic guide was able to answer your questions about the Limbo project and expect to hear from you or from your own Limbo sound design.

Further Help and Learning Resources

To learn more about Wwise, please refer to:

- *Wwise Project Adventure* document and companion project
- The Wwise video tutorials on the Audiokinetic YouTube Channel at: http://www.youtube.com/user/AudiokineticWwise
- The Wwise user guide and online help provided with the Wwise authoring application.