SYNTHETIC LIFE
CORPORATION
Liliana Darwinian

[LD] Ambiance Sounds Manual

Greetings and thank you for your interest in this product!

Ambiance Sounds reproduces environmental sounds so as to mimic several kinds of environments by playing different sounds according the time of day, that is SL four-hour day.

Figure 1 - Ambiance Sounds

Summary

FEATURES

- Menu driven
- Use your own sounds
- Simulate any environment
- Programmable via notecard
- Random occasional sounds
- Continuous sounds
- Define probability and volume for each sound
- Define at what times of day the sounds will play
- Each Sound Engine can play as many sounds as you like
- Define as many Sound Engines as you like to simulate different sets in an environment
- Contains two scripts that can be put in any modifiable object
- Contains one usage example disguised as a stone that simulates a beach & woods environment
- Free Updates

USING AMBIANCE SOUNDS

The Ambiance Sounds system consists of two scripts: the Sound Engine and a Menu that controls the engine(s). You can add the scripts to any modifiable object to be used as a disguise, like a rock, a lamp, whatever. You can add as many Sound Engines as you wish in a single object so as to reproduce several kinds of ambiance sounds under different circumstances. For example, the "Example Usage - Ambiance Sounds Rock" provided with the product contains two sound engines, one generates a continous sea sound with occasional waves sounds, while the other generates other occasional ambiance sounds depending on time of day, birds singing during the day, owls, crickets and some distant barks during the night, maybe a rooster in the early morning and so on. Each sound has a probability of being played thus making a more realistic simulation.

The programming, so to say, of the sound system is done in a configuration notecard where you specify all sounds to be played, when, how and where. There must be a Config notecard for each Sound Engine. You don't need to rename them, all you have to do is drag & drop each one in turn into the object's inventory**.**

Step by Step

1) Edit the object to be used as a disguise and select the Contents tab.

2) Drag & drop the Menu script into the object's inventory.

3) Create a notecard and name it "Config" (without the quotes).

4) Edit the notecard and specify the sounds to be played using the syntax explained below. Save the notecard.

5) Drag & drop the Config notecard into the object's inventory.

6) Drag & drop the Sound Engine script into the object's inventory.

Repeat steps 3 through 6 as many times as you wish.

Notes

- When you drag & drop a second notecard named "Config" it will be automatically renamed to "Config 1". And so on.

- When you drag & drop a second "Sound Engine nn.nn #" it will be automatically renamed to "Sound Engine nn.nn # 1". And so on.

- Script "Sound Engine nn.nn #" will use notecard "Config", script "Sound Engine nn.nn # 1" will use notecard "Config 1", and so on.

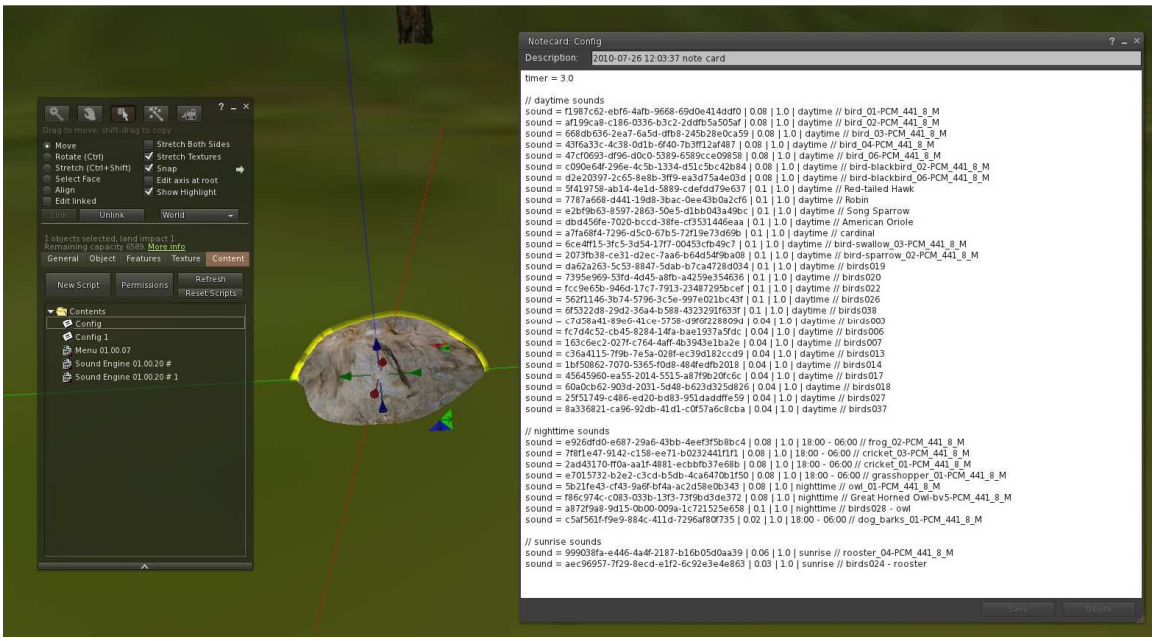


Figure 2 - Editting Config File

CONFIG NOTECARD COMMANDS

//

Double slashes indicate that a comment follows. Comments are ignored by the sound engine.

E.g.:

// this is a comment

SOUND

Defines a sound file. More often used for defining occasional, random sounds, like chirping birds, barking dogs.

The same file can be defined many times allowing you to have for example a dog that barks more often at night.

Syntax:

sound = filename | probability | volume | period_of_the_day | top_north_east | bottom_south_west

or

sound = filename | probability | volume | start_sec - end_sec | top_north_east | bottom_south_west

or

sound = filename | probability | volume | start_time - end_time | top_north_east | bottom_south_west

where:

filename: name of sound file or UUID

probability: probability of sound being played in a time cycle (0.0 - 1.0)

volume: sound volume (0.0 - 1.0)

period_of_the_day: daytime, nighttime, morning, afternoon, all day, sunrise, sunset

start_sec: start time (in seconds)

end_sec: end time (in seconds)

start_time: start time (in hours & minutes, format: hh:mm, 24h format/military time)

end_time: end time (in hours & minutes, format: hh:mm, 24h format/military time)

top_north_east: top (north-east) position in region coordinates (vector) (this parameter is optional)

bottom_south_west: bottom (south-west) position in region coordinates (vector) (this parameter is optional)

period	seconds start	seconds end	time start	time end
daytime	1801	12600	06:00	18:00
nighttime	12601	1800	18:00	06:00
morning	1801	7200	06:00	12:00
afternoon	7201	12600	12:00	18:00
all day	0	14400	00:00	24:00
sunrise	1500	2700	05:00	07:00
sunset	6300	7500	17:00	19:00

Notes

- if you define a sound file using a file name, the specified file must be in the object's inventory; the device will warn you if the file is not in the object's inventory.

- If you define a sound file using a UUID, the file does not need to be in the object's directory; in this case the device cannot verify the existence and validity of the specified UUID and an error will be generated if the sound file does not exist or it's not a sound file.

- Times of day will be converted internaly to corresponding times of the Second Life 4-hour day cycle.

- If you specify both top_north_east and bottom_south_west parameters the specfied sound will be heard only within the specified region coordinates.

E.g.:

sound = crickets | 0.02 | 1.0 | nightime

sound = Bird1 | 0.1 | 1.0 | daytime

sound = frog1 | 0.02 | 1.0 | 18:00 - 06:00

sound = machine | 0.2 | 1.0 | all day | <209.0, 181.0, 26.0> | <167.0, 142.0, 50.0>

LOOP

Defines a soundfile to be played in a continual loop.

This is useful for defining sounds that play constantly like waves, for example.

Syntax:

sound = filename | probability | volume

where:

filename: sound file or UUID

probability: probability of sound being stopped in a time cycle (0.0 - 1.0)

volume: sound volume (0.0 - 1.0)

Note

- If you want the sound to play continually then probability should be zero.

E.g.:

loop = waves_sound | 0.0 | 0.8

TIMER

Defines timer interval. Default is 4 seconds, mininum is 2 seconds. Timer can be defined only once per engine.

Realistic environments can be simulated by carefuly defining Timer and Probability values. Don't forget that each engine uses its own timer.

Syntax:

timer = <time interval in seconds>

E.g.:

timer = 5.0

Samples

Here you have Config notecards used for the usage example provided:

Config

```
timer = 3.0

// daytime sounds
sound = f1987c62-ebf6-4afb-9668-69d0e414ddf0 | 0.08 | 1.0 | daytime //
bird_01-PCM_441_8_M
sound = af199ca8-c186-0336-b3c2-2ddfb5a505af | 0.08 | 1.0 | daytime //
bird_02-PCM_441_8_M
sound = 668db636-2ea7-6a5d-dfb8-245b28e0ca59 | 0.08 | 1.0 | daytime //
bird_03-PCM_441_8_M
sound = 43f6a33c-4c38-0d1b-6f40-7b3ff12af487 | 0.08 | 1.0 | daytime //
bird_04-PCM_441_8_M
sound = 47cf0693-df96-d0c0-5389-6589cce09858 | 0.08 | 1.0 | daytime //
bird_06-PCM_441_8_M
sound = c090e64f-296e-4c5b-1334-d51c5bc42b84 | 0.08 | 1.0 | daytime //
bird-blackbird_02-PCM_441_8_M
sound = d2e20397-2c65-8e8b-3ff9-ea3d75a4e03d | 0.08 | 1.0 | daytime //
bird-blackbird_06-PCM_441_8_M
sound = 5f419758-ab14-4e1d-5889-cdefdd79e637 | 0.1 | 1.0 | daytime //
Red-tailed Hawk
sound = 7787a668-d441-19d8-3bac-0ee43b0a2cf6 | 0.1 | 1.0 | daytime //
Robin
sound = e2bf9b63-8597-2863-50e5-d1bb043a49bc | 0.1 | 1.0 | daytime //
Song Sparrow
sound = dbd456fe-7020-bccd-38fe-cf3531446eaa | 0.1 | 1.0 | daytime //
American Oriole
sound = a7fa68f4-7296-d5c0-67b5-72f19e73d69b | 0.1 | 1.0 | daytime //
cardinal
sound = 6ce4ff15-3fc5-3d54-17f7-00453cfb49c7 | 0.1 | 1.0 | daytime //
bird-swallow_03-PCM_441_8_M
sound = 2073fb38-ce31-d2ec-7aa6-b64d54f9ba08 | 0.1 | 1.0 | daytime //
bird-sparrow_02-PCM_441_8_M
sound = da62a263-5c53-8847-5dab-b7ca4728d034 | 0.1 | 1.0 | daytime //
birds019
sound = 7395e969-53fd-4d45-a8fb-a4259e354636 | 0.1 | 1.0 | daytime //
birds020
sound = fcc9e65b-946d-17c7-7913-23487295bcef | 0.1 | 1.0 | daytime //
birds022
sound = 562f1146-3b74-5796-3c5e-997e021bc43f | 0.1 | 1.0 | daytime //
birds026
sound = 6f5322d8-29d2-36a4-b588-4323291f633f | 0.1 | 1.0 | daytime //
birds038
sound = c7d58a41-89e6-41ce-5758-d9f6f228809d | 0.04 | 1.0 | daytime //
birds003
sound = fc7d4c52-cb45-8284-14fa-bae1937a5fdc | 0.04 | 1.0 | daytime //
birds006
sound = 163c6ec2-027f-c764-4aff-4b3943e1ba2e | 0.04 | 1.0 | daytime //
birds007
sound = c36a4115-7f9b-7e5a-028f-ec39d182ccd9 | 0.04 | 1.0 | daytime //
birds013
```

```
sound = 1bf50862-7070-5365-f0d8-484fedfb2018 | 0.04 | 1.0 | daytime //
birds014
sound = 45645960-ea55-2014-5515-a87f9b20fc6c | 0.04 | 1.0 | daytime //
birds017
sound = 60a0cb62-903d-2031-5d48-b623d325d826 | 0.04 | 1.0 | daytime //
birds018
sound = 25f51749-c486-ed20-bd83-951daddffe59 | 0.04 | 1.0 | daytime //
birds027
sound = 8a336821-ca96-92db-41d1-c0f57a6c8cba | 0.04 | 1.0 | daytime //
birds037

// nighttime sounds
sound = e926dfd0-e687-29a6-43bb-4eef3f5b8bc4 | 0.08 | 1.0 | 18:00 - 06:00
// frog_02-PCM_441_8_M
sound = 7f8f1e47-9142-c158-ee71-b0232441f1f1 | 0.08 | 1.0 | 18:00 - 06:00
// cricket_03-PCM_441_8_M
sound = 2ad43170-ff0a-aa1f-4881-ecbbfb37e68b | 0.08 | 1.0 | 18:00 - 06:00
// cricket_01-PCM_441_8_M
sound = e7015732-b2e2-c3cd-b5db-4ca6470b1f50 | 0.08 | 1.0 | 18:00 - 06:00
// grasshopper_01-PCM_441_8_M
sound = 5b21fe43-cf43-9a6f-bf4a-ac2d58e0b343 | 0.08 | 1.0 | nighttime //
owl_01-PCM_441_8_M
sound = f86c974c-c083-033b-13f3-73f9bd3de372 | 0.08 | 1.0 | nighttime //
Great Horned Owl-bv5-PCM_441_8_M
sound = a872f9a8-9d15-0b00-009a-1c721525e658 | 0.1 | 1.0 | nighttime //
birds028 - owl
sound = c5af561f-f9e9-884c-411d-7296af80f735 | 0.02 | 1.0 | 18:00 - 06:00
// dog_barks_01-PCM_441_8_M

// sunrise sounds
sound = 999038fa-e446-4a4f-2187-b16b05d0aa39 | 0.06 | 1.0 | sunrise //
rooster_04-PCM_441_8_M
sound = aec96957-7f29-8ecd-e1f2-6c92e3e4e863 | 0.03 | 1.0 | sunrise //
birds024 - rooster
```

Config 1

```
timer = 3.0

// beach: constant background sound + occasional random strong waves
loop = 46b72f72-0bb1-34dc-6bf0-92e7515878c2 | 0.0 | 0.8 // ocean-regular-
waves-light_02b-PCM_441_8_M
sound = 35f07187-d4f4-69c3-e6a8-649b876bdfda | 0.05 | 0.8 | all day //
ocean-regular-waves-light_03b-PCM_441_8_M
sound = b60ae680-57ae-ac22-3f4f-81c681789521 | 0.05 | 0.8 | all day //
ocean-wave-med_04-PCM_441_8_M
sound = c61ddb5e-a548-855e-9338-7f3b96b8438e | 0.05 | 0.8 | all day //
ocean-wave-med_05-PCM_441_8_M
sound = 19317795-9ce8-5729-acc1-87d0b7ac245f | 0.05 | 0.8 | all day //
ocean-wave-med_06-PCM_441_8_M
```

UPLOADING YOUR OWN SOUNDS

You can record yourself or get some free sounds from the Internet. Whatever the source you may need to convert the sound files to the Second Life standard:

Max Length: 10 s

Format: WAV - PCM Uncompressed

Attributes: 44100 Hz , 8 bits , Stereo/Mono

ACCESS RIGHTS

Only owner can operate this device.

MENU

[On]

Turn on all Sound Engines.

[Off]

Turn off all Sound Engines.

[Status]

Display status and free memory for each Sound Engine.

[Reset]

Reset all scripts.

[Done]

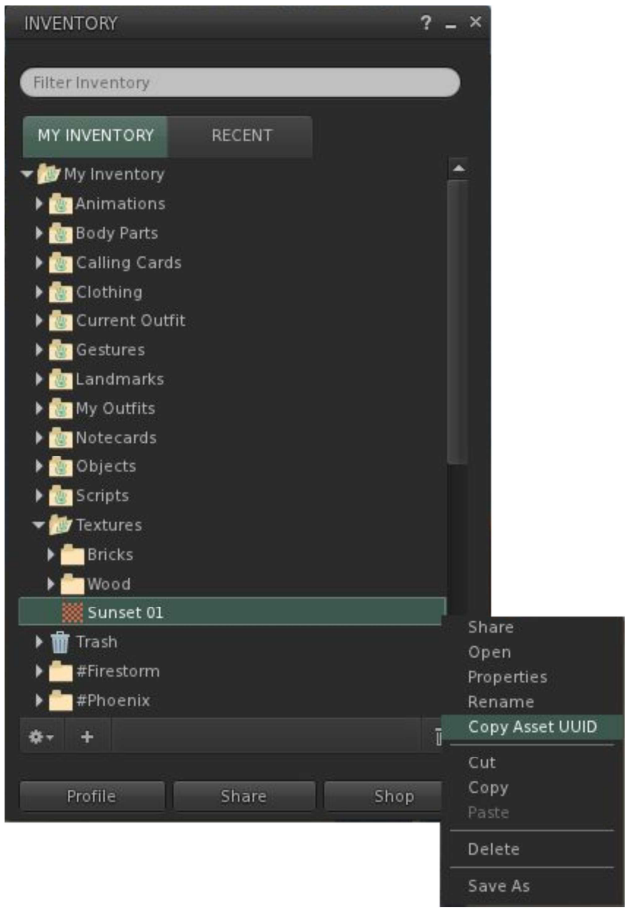
Close menu.

NOTE: In order to minimize lag the menu has a timeout of 60 seconds.

FAQ

What's UUID?

UUID is an abbreviation for Universally Unique Identifier. It is a 128-bit (16 byte) value which is generated in such a way as to make collisions very unlikely. They are often represented as a string of 32 hex characters with four dashes interspersed. In Second Life every object and every avatar has an UUID.



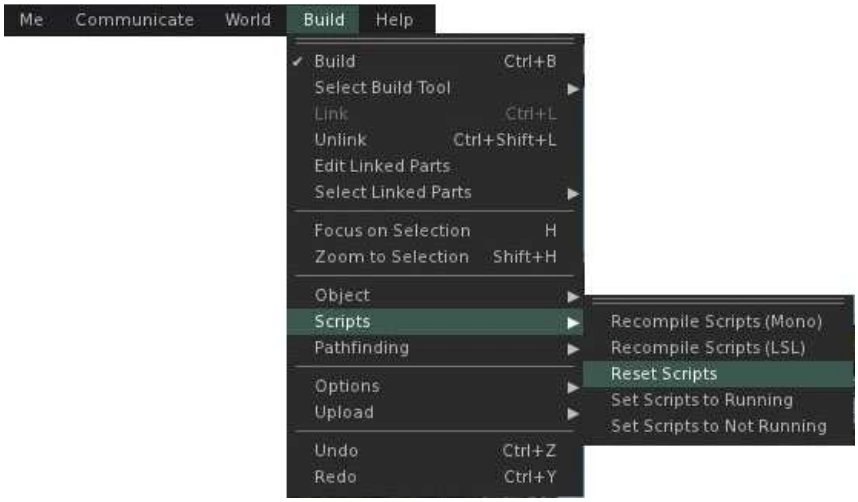
Right-click texture and select **Copy Asset UUID** then paste it in the notecard

Figure 3 - Getting the UUID of a Texture

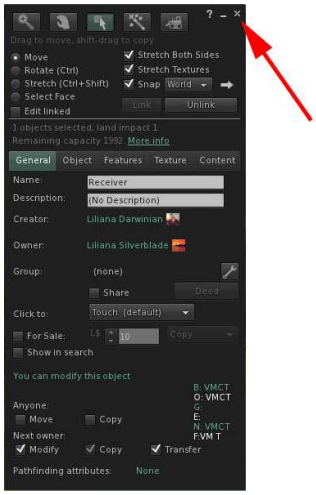
How do I Reset the Scripts?

In most situations you can reset the scripts using the object's main menu (Options > Reset), this is called a "soft reset". On rare occasions though you may need to perform a so called "hard reset":

1) Right-click the object and select **Edit**



2) Open the Build Menu, select Scripts, then **Reset Scripts**



3) Close the Edit Window

Figure 4 - Resetting Scripts (Hard Reset)

Repeat the operation if necessary.

I don't hear anything!

- check your volume settings.
- make sure the sound file exists in the object's inventory.
- if you are using UUID make sure you got the right one.
- check the probability of the sound.

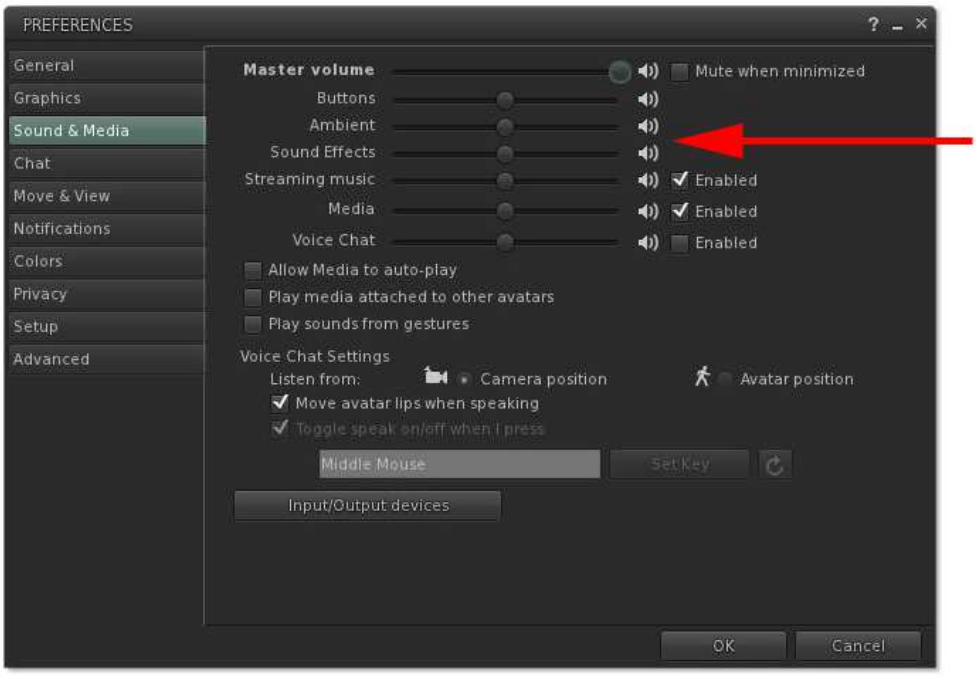


Figure 5 - Volume Settings

How do I check for updates?

- Rez the [LD] Updater nn.nn
- Click on it and select Update

PERMISSIONS

Prims		Scripts	
			
☑	Modify		No Modify
☑	Copy	☑	Copy
	No Transfer		No Transfer

Sold as is and without refund. However, I will fix and replace your book if defective. Because no-transfer items cannot be returned, we cannot offer refunds, however if this product does not fit your needs, please contact us and let us know why so that we can improve our products.
Due to the nature of SL permission system and numerous cheat cases all sales are final: no exchange and no refund.
More details on our website.
http://syntheticlifecorporation.wordpress.com/policies/

Troubleshooting

• Failed Delivery
- Please DO NOT re-purchase an item if your transaction fails or something goes wrong without contacting us first. WE CAN RE-SEND ITEMS.
- You may need to re-log if you don't see your purchase in your inventory. If it's not there, send me a notecard with the date, time, location of your purchase and a copy of transaction history.
- Because no-transfer items cannot be returned, we cannot offer refunds except on double purchases. Please drop a notecard with the transaction history of the double purchase.

• Inventory Loss
Please take a look here:
http://wiki.secondlife.com/wiki/Inventory_loss

Contact

Questions? comments? suggestions? bug reports? Contact us inworld:

If you have any questions or comments on this product please feel free to contact me any time.
I am online everyday and will respond as soon as possible.

I can be reached through IM or notecards but notecards are preferred because IM's often get capped.

Feedback is always appreciated, good or bad.
Your input is important and can help in the improvement of this product and the development of future products.

If you think you have found a bug then please fill up a bug report providing as much information as possible about the error and how to reproduce it and send it to me.

LILIANA DARWINIAN

secondlife:///app/agent/37d0c09d-f157-4e7e-92f9-88092d2cf469/about
https://my.secondlife.com/liliana.darwinian
http://world.secondlife.com/resident/37d0c09d-f157-4e7e-92f9-88092d2cf469

★ **PLEASE NOTE:**
IMs usually get capped so please send a notecard and include a copy of transaction history if necessary.

• **Email**
Liliana.Darwinian@gmail.com

• **Blog**
http://syntheticlifecorporation.wordpress.com/

• **Marketplace Store**
https://marketplace.secondlife.com/stores/114579

Legal Blabla

Liliana Darwinian