Animator Sounds

VS Visualizer can play recorded sound samples to produce more complete and realistic presentations of animated tests. Sounds are implemented under the OpenAL standard. Sounds are typically added to animations to represent engine noise, wind noise, and tire noises, but any appropriate sound can be used. Individual sound sample files in `.wav` or `.ogg` format are specified on an **Animator: Sound Sample** screen. Each sample can be looped, and its volume (loudness) and frequency (pitch) can be adjusted dynamically during playback, or individual sounds can be triggered to play when some specific action or event occurs using an **Animator: Sound Modifier** dataset. Several sounds naturally associated with a single object can be grouped together on an **Animator: Sound Set** screen.

Animator: Sound Sample

This is the main screen for specifying an individual sound and the way it is played. Each sound consists of a recorded sample stored in a file in `.wav` or `.ogg` format.

Discussion

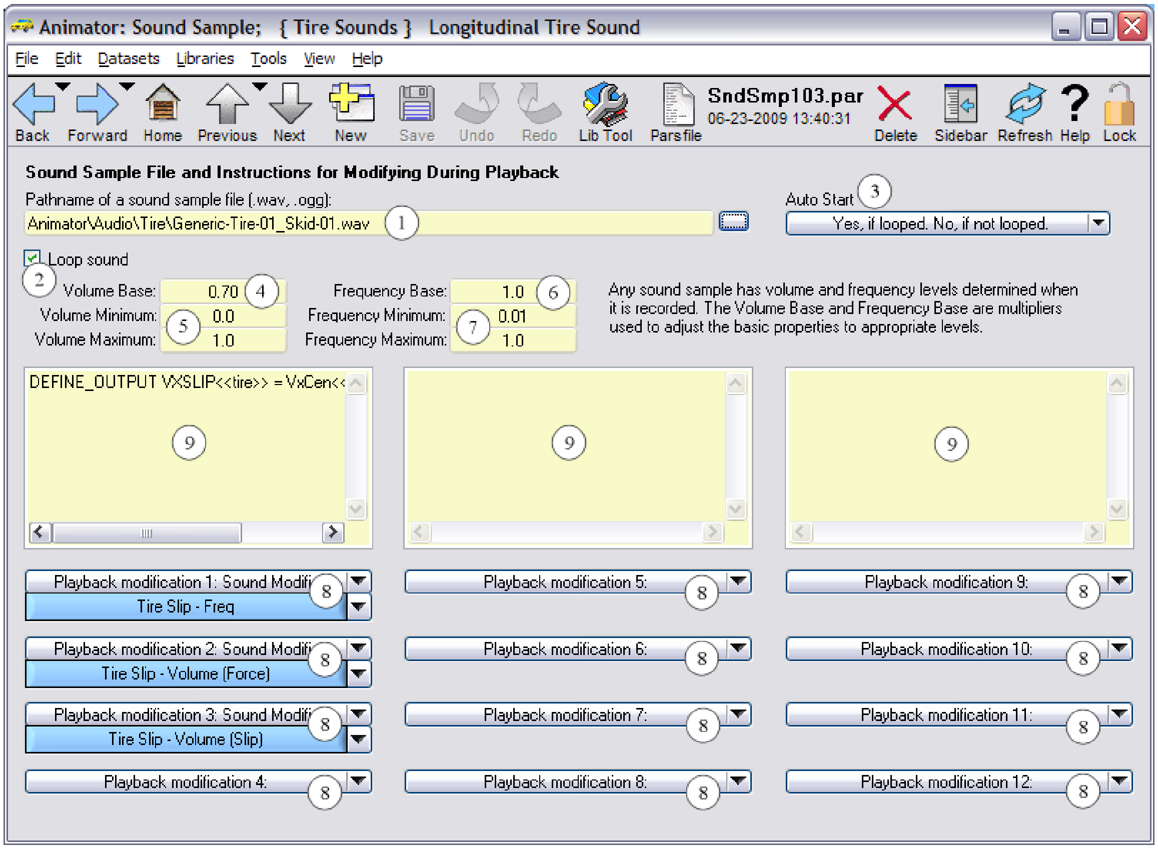
Continuous sounds or sounds with long duration can be produced by looping the playback of short samples. The playback of looped sounds normally starts at the beginning of the animation and is simply modified throughout the animation for loudness and pitch.

Some sounds aren’t looped and must be started at specific times. For example, a tire might hit a puddle and cause a splashing sound. The playback of the splash should begin at the moment the tire hits the puddle. Sounds like this are controlled by a trigger. Triggers are discussed in more detail in the **Animator: Sound Modifier** section.

Each recorded sample has some volume (loudness) level and some frequency (pitch) range based on the nature of the sound when it was recorded. Volume and pitch can be modified by applying multipliers. The value of a multiplier is the product of a *base multiplier* and possibly one or more optional *dynamic multipliers*.

The base multiplier is used to set the basic level of a sound. Different samples might vary widely in their recorded volume, for example, and need to be adjusted to work correctly with all the other sounds in use. A base frequency multiplier might be used to adjust an engine sound for playback representing an RPM range different from that of the recorded sample.

Dynamic multipliers change the way sounds are played back as the animation progresses, based on the calculated values of any output variables. They are discussed in more detail in the **Animator: Sound Modifier** section.



Animator: Sound Sample; { Tire Sounds } Longitudinal Tire Sound

SndSmp103.par
06-23-2009 13:40:31

Sound Sample File and Instructions for Modifying During Playback

Pathname of a sound sample file (.wav, .ogg): Animator\Audio\Tire\Generic-Tire-01_Skid-01.wav

Auto Start: Yes, if looped. No, if not looped.

☑ Loop sound

Volume Base: 0.70
Volume Minimum: 0.0
Volume Maximum: 1.0

Frequency Base: 1.0
Frequency Minimum: 0.01
Frequency Maximum: 1.0

Any sound sample has volume and frequency levels determined when it is recorded. The Volume Base and Frequency Base are multipliers used to adjust the basic properties to appropriate levels.

DEFINE_OUTPUT VXSLIP<<tire>> = VxCen<<

Playback modification 1: Sound Modifi
Tire Slip - Freq

Playback modification 2: Sound Modifi
Tire Slip - Volume (Force)

Playback modification 3: Sound Modifi
Tire Slip - Volume (Slip)

Playback modification 4:

Playback modification 5:

Playback modification 6:

Playback modification 7:

Playback modification 8:

Playback modification 9:

Playback modification 10:

Playback modification 11:

Playback modification 12:

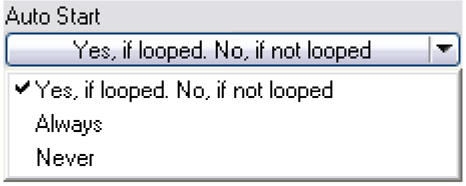
Figure 1: The ***Animator: Sound Sample*** *screen*

User Settings

① Pathname to an individual sound sample file (keyword = `ADD_3D_SOUND`). Type or paste the name of the file containing the sound sample, or use the browse button to locate it.

② Checkbox to control whether the sound sample is looped. Looping enables short samples of sounds to represent long or continuous sounds.

③ Drop-down list to control whether playback of a sound is automatically started when the animation starts.


Auto Start
Yes, if looped. No, if not looped
✔ Yes, if looped. No, if not looped
Always
Never

The first option is "Yes, if looped. No, if not looped.", and is the default. It works for most applications. It starts playback of the selected sound as soon as the animation starts if the "Loop sound" box is checked and does not if the box is not checked. A sound that doesn't auto start must be started with a trigger.

The second option, “Always,” causes the sound to auto start whether looping is selected or not, and the third option, “Never,” disables auto start for this sound, so it always requires a trigger.

④ Base volume multiplier for the selected sound (keyword = `SOUND_VOL_MULT`). This setting is used to adjust the basic playback volume level of a sound. It is multiplied by any dynamic multipliers to determine the playback level at any instant.

⑤ Minimum and maximum volume multipliers for the selected sound (keywords = `SOUND_VOL_MIN` and `SOUND_VOL_MAX`). Since the base volume setting and any dynamic modifiers for a sound are multiplied together to obtain the current net multiplier value, it is possible to generate multipliers outside a reasonable range. Use these fields to limit the extremes of the product of the base multiplier and all dynamic multipliers.

⑥ Base frequency multiplier for the selected sound (keyword = `SOUND_FREQ_MULT`). This setting is used to adjust the basic playback pitch of a sound. It is multiplied by any dynamic multiplier to determine the playback pitch at any instant. The implementation of frequency scaling causes the playback length of the sample to be adjusted in inverse proportion to the frequency multiplier. For example, doubling the playback frequency reduces the duration of the sound by half.

⑦ Minimum and maximum frequency multipliers for the selected sound (keywords = `SOUND_FREQ_MIN` and `SOUND_FREQ_MAX`). Since the base frequency setting and any dynamic modifiers for a sound are multiplied together to obtain the current net multiplier value, it is possible to generate multipliers outside a reasonable range. Use these fields to limit the extremes of the product of the base multiplier and all dynamic multipliers.

Note Some sound cards process audio information using log functions and may generate an error if the value of the frequency multiplier is zero. If this occurs, set the minimum value if the multiplier to a very small value (e.g., 0.001).

⑧ Links to **Animator: Sound Modifier** data sets. Each modifier defines a multiplier or trigger for a sound. A sound may be multiplied according to several specifications. For example, the volume of an engine sound may be increased with increasing load (throttle) and with increasing RPM. Multiple triggers may be defined to play a sound sample in response to a number of different actions or conditions. A total of twelve modifiers can be specified for any sound.

⑨ Miscellaneous data fields that can be used to provide information that is needed by the set of modifier specifications that follow the field. For example, VS commands might be used to define an output variable to be used by a group of modifiers. The effects of the three miscellaneous data fields are interleaved with the twelve links to modifier data so that leftmost data field is read, followed by the links labeled “Playback modification 1” through “Playback modification 4”, the center data field is then read followed by the links labeled “Playback modification 5” through “Playback modification 8”, then the rightmost field is read followed by the links labeled “Playback modification 9” through “Playback modification 12”. This means that information in the left data field can be used by all twelve links, data in the center field can be used by the links labeled “Playback modification 5” through “Playback modification 12”, and data in the right field can be used by the links labeled “Playback modification 9” through “Playback modification 12”.

Animator: Sound Modifier

This screen (Figure 3) configures a single modifier for a sound.

Discussion

Each sound can be modified by volume scaling, frequency scaling, or triggered to turn on and off. Up to twelve modifiers can be specified for each sound. All the modifiers that alter a sound are linked to the sound sample on the **Animator: Sound Sample** screen (Figure 1).

The value of each modifier is the output of a simple table function. The input to the table function (the independent variable) is the value of a VehicleSim output variable (either an internally defined variable or one created with VS Commands). The output is a modifier to be included in the handling of the sound. The modifier can be defined to be a multiplier or trigger.

The table is defined by four pairs of values. Each pair consists of an input value and a modifier value, and represents one point in the table. The inputs must be defined in increasing order from “Point 1” to “Point 4”. The table is linear interpolated between points and flat-line extrapolated beyond the range of the points.

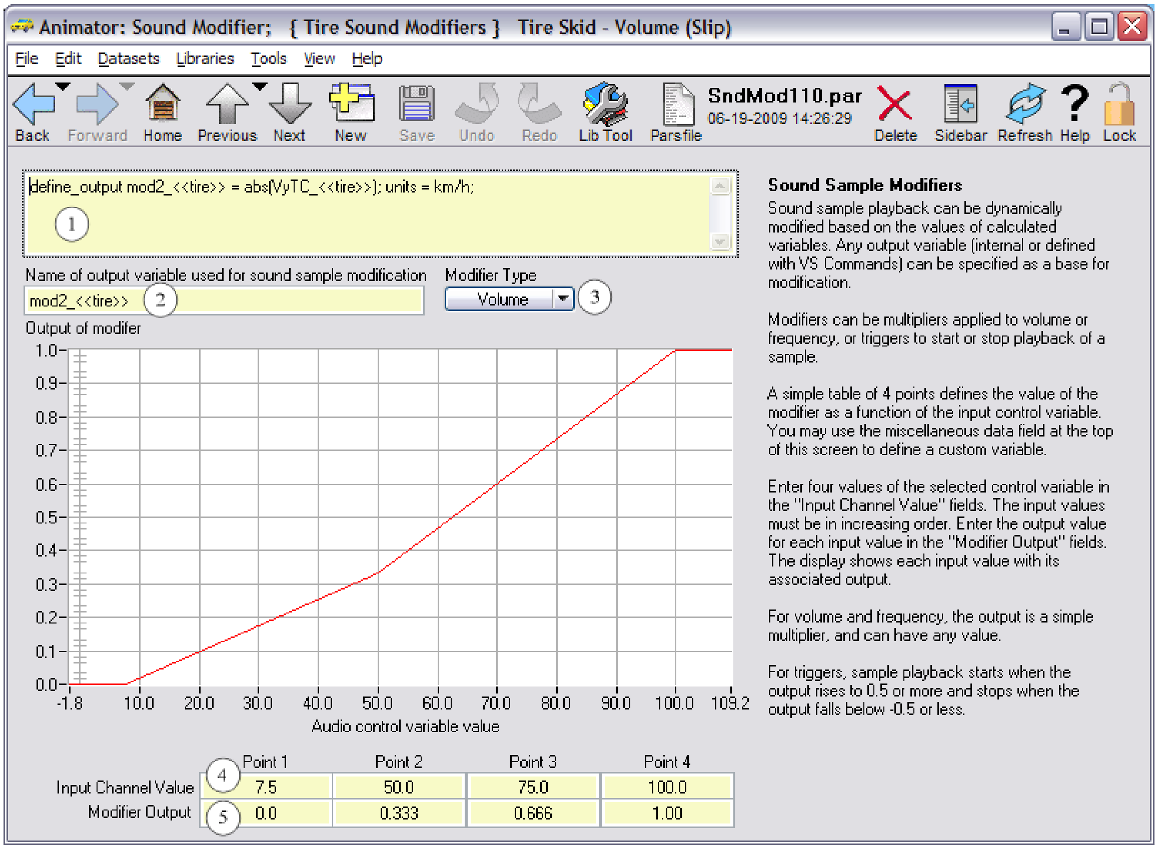
The net sound volume multiplier at any moment is the product of all the volume multipliers linked to a particular sound; the net frequency multiplier is the product of all the frequency multipliers linked to a particular sound. Both multipliers are subject to limits set as minimum and maximum values on the **Animator: Sound Sample** screen. The interpretation of negative values passed to a multiplier is not defined under the OpenAL standard. As a result, their effects are hardware/driver dependent and unpredictable, so use of negative multipliers is not recommended.

Triggers are used to control the ON/OFF state of playback of a sound. Any value, positive or negative, can be passed to a trigger. However, only two values cause an action. Data passed to a trigger is interpreted as follows:

- If the output of the trigger is less than +0.5 and rises to be greater than or equal to +0.5, the sound is turned ON if it is presently OFF. If it is already ON, no modification takes place; it stays ON.
- If the output of the trigger is greater than -0.5 and falls to be less than or equal to -0.5, the sound is turned OFF if it is presently ON. If it is already OFF, no modification takes place; it stays OFF.

Triggers are mainly used to control the playback of sound samples that have time-related content. Consider for example a bell that is rung and then the sound dies away slowly. If the sound were looped and controlled only by volume multipliers, it might become audible after the initial ringing as the sound is already dying away. The sound probably wouldn’t make sense in context.

Triggers are often used to control short-duration sounds related to some specific occurrence or location.



define_output mod2_<<tire>> = abs(VyTC_<<tire>>); units = km/h;

Name of output variable used for sound sample modification: mod2_<<tire>>

Modifier Type: Volume

	Point 1	Point 2	Point 3	Point 4
Input Channel Value	7.5	50.0	75.0	100.0
Modifier Output	0.0	0.333	0.666	1.00

Sound Sample Modifiers

Sound sample playback can be dynamically modified based on the values of calculated variables. Any output variable (internal or defined with VS Commands) can be specified as a base for modification.

Modifiers can be multipliers applied to volume or frequency, or triggers to start or stop playback of a sample.

A simple table of 4 points defines the value of the modifier as a function of the input control variable. You may use the miscellaneous data field at the top of this screen to define a custom variable.

Enter four values of the selected control variable in the "Input Channel Value" fields. The input values must be in increasing order. Enter the output value for each input value in the "Modifier Output" fields. The display shows each input value with its associated output.

For volume and frequency, the output is a simple multiplier, and can have any value.

For triggers, sample playback starts when the output rises to 0.5 or more and stops when the output falls below -0.5 or less.

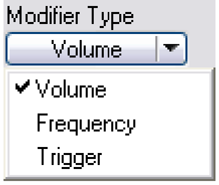
*Figure 2: The **Animator: Sound Modifier** screen*

User Settings

① Miscellaneous data field to provide information need for this modifier. This field is mainly used to create an output variable using VS Commands. Sound volume scaling, for example, might use the absolute value of an internal variable.

② Name of an output variable used as input to the modifier table (keyword = `SOUNDMOD_INPUT_NAME`). This must be a valid output variable, either internal or user-created.

③ Drop-down list to select the modifier type. Three types of modifiers are available: volume, frequency, and trigger.



When “Volume” is selected, the output of the modifier is multiplied by all other volume multipliers defined for the sound under which it is linked to obtain the net multiplier for amplification of the sound. When “Frequency” is selected, the output of the modifier is multiplied by all other frequency multipliers defined for the sound under which it is linked to obtain the net multiplier for adjusting the pitch of the sound. When “Trigger” is selected the modifier is used to control the ON/OFF state of playback of the sound.

④ List of values of the input channel to be associated with values of the modifier. The points must be defined with input values in increasing order from Point 1 to Point 4 (left to right). The table is linearly interpolated between points and flat line extrapolated outside their range.

⑤ List of values of the modifier associated with specific values of the input channel. For volume and frequency modifiers, these should always be positive. The table is linearly interpolated between points and flat line extrapolated outside their range.

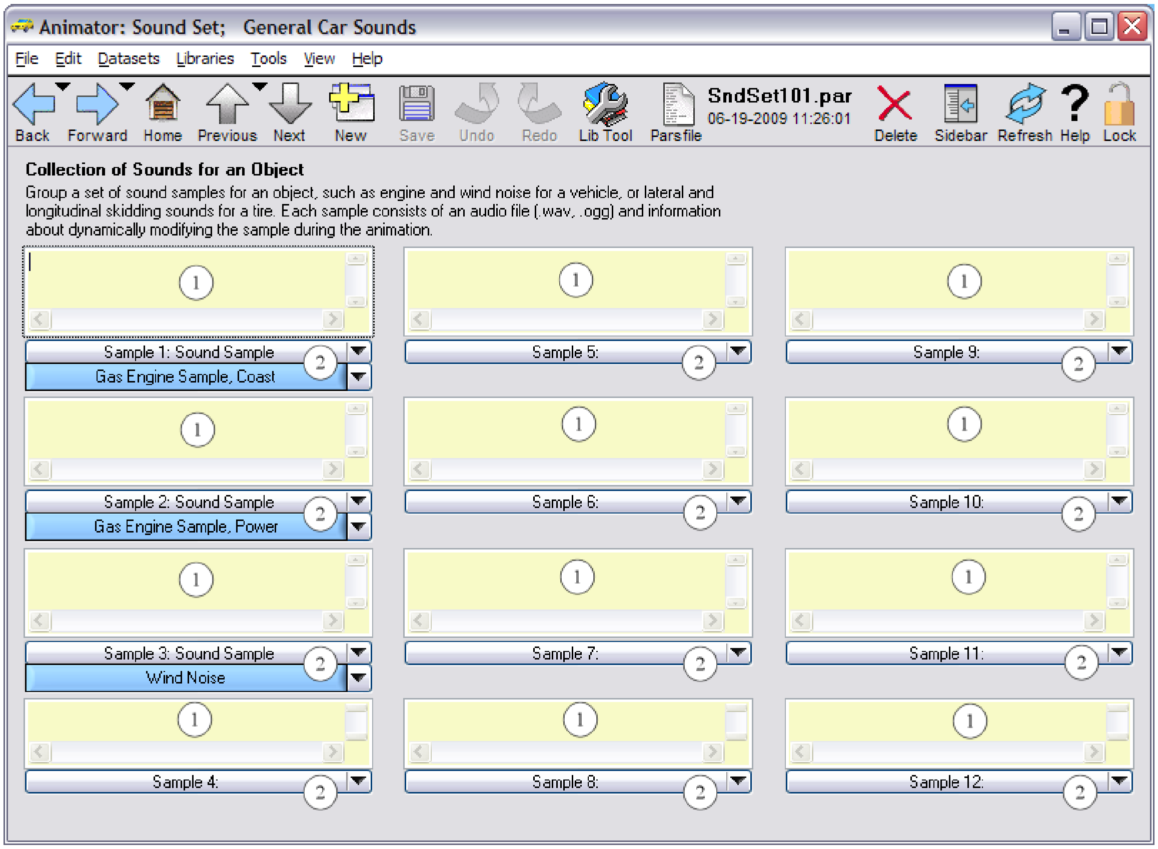
Animator: Sound Set

Use this screen to group together all the sound specifications that are naturally associated with an object.

Discussion

The animator can play recorded sound samples to produce more complete and realistic presentations of animated test. Sounds are typically added to animations to represent engine noise, wind noise, and tire noises, but any appropriate sound can be used. Each sound is associated with the origin of an animator reference frame, so any sounds that move with one reference frame should be grouped together. For example, engine sounds are naturally grouped with (and move with) a vehicle and are linked with the vehicle animation shapes. Tire sounds move with the tire and are linked with the tire. In each case, the **Animator: Sound Set** screen is used to group all the sounds to be linked to one reference frame. Up to twelve different sounds can be linked to one Sound Set data set. If more sounds are needed for one object, use a **Generic Data Group** screen.

> **Note** The documentation for the screen **Animator: Reference Frame** describes how you define reference frames.



Collection of Sounds for an Object

Group a set of sound samples for an object, such as engine and wind noise for a vehicle, or lateral and longitudinal skidding sounds for a tire. Each sample consists of an audio file (.wav, .ogg) and information about dynamically modifying the sample during the animation.

Sample 1: Sound Sample
Gas Engine Sample, Coast

Sample 2: Sound Sample
Gas Engine Sample, Power

Sample 3: Sound Sample
Wind Noise

Sample 4:

Sample 5:

Sample 6:

Sample 7:

Sample 8:

Sample 9:

Sample 10:

Sample 11:

Sample 12:

*Figure 3: The **Animator Sound Set** screen*

User Settings

Figure 3 shows the **Animator: Sound Set** screen with three sounds linked together for a vehicle.

Up to twelve different sounds can be linked to one Sound Set data set. If more sounds are needed for one object, use a **Generic Data Group** screen to link more than one **Animator: Sound Set** screen.

① Use this miscellaneous data field to specify information needed by the sound sample, such as VS Commands defining output variables.

② Link to an **Animator: Sound Sample** dataset.