orxSOUND structure

Summary

```
[SoundTemplate]
Sound        = path/to/SoundFile
Music        = path/to/MusicFile
Bus          = <string>
Loop         = <bool>
Pitch        = <float>
Volume       = <float>
Panning      = <float>
Mix          = <bool>
DistanceList = <float> (# <Float>)
GainList     = <float> (# <float>)
RollOff      = <float>
FilterList   = FilterTemplate1 # ... # FilterTemplateN
KeepInCache = <bool>
```

BiquadFilter

```
[BiquadFilter]
Type           = biquad
UseCustomParam = <bool>
A0             = <float>
A1             = <float>
A2             = <float>
B0             = <float>
B1             = <float>
B2             = <float>
```

LowPassFilter

```
[LowPassFilter]
Type           = lowpass
UseCustomParam = <bool>
Frequency      = <float>
Order          = <int>
```

HighPassFilter

```
[HighPassFilter]
Type           = highpass
UseCustomParam = <bool>
Frequency      = <float>
```

```
Order           = <int>
```

BandPassFilter

```
[BandPassFilter]
Type            = bandpass
UseCustomParam  = <bool>
Frequency       = <float>
Order           = <int>
```

LowShelfFilter

```
[LowShelfFilter]
Type            = lowshelf
UseCustomParam  = <bool>
Frequency       = <float>
Q               = <float>
Gain            = <float>
```

HighShelfFilter

```
[HighShelfFilter]
Type            = highshelf
UseCustomParam  = <bool>
Frequency       = <float>
Q               = <float>
Gain            = <float>
```

NotchFilter

```
[NotchFilter]
Type            = notch
UseCustomParam  = <bool>
Frequency       = <float>
Q               = <float>
```

PeakingFilter

```
[PeakingFilter]
Type            = peaking
UseCustomParam  = <bool>
Frequency       = <float>
Q               = <float>
```

```
Gain            = <float>
```

DelayFilter

```
[DelayFilter]
Type            = delay
UseCustomParam  = <bool>
Delay           = <float>
Decay           = <float>
```

Details

Here's a list of the available properties for an `orxSOUND` structure:

- `Sound`: Specifies which sound file to use. The whole file will be loaded in memory before being played. This should be used for short sounds only (not for buffered streamed musics).
- `Music`: If no `Sound` is specified, this will be used as a path to a music file. Music differs from sounds mainly in the fact they're streamed from disk and buffered in memory at runtime to prevent having big files completely loaded in memory.
- `Bus`: Assign the sound/music to a named bus. If this is not specified, the default `Master` bus will be used.
- `Loop`: Specifies if the sound should be played in loop. Its default value is `false`.
- `Pitch`: Defines a pitch value for the playing sound (ie. a relative frequency). Its default value is 1.0 which means the sound will be played at the same frequency it was recorded. 2.0 will be a pitch twice as high, 0.5 is a pitch twice as low. 0.0 has no meaning and is ignored.
- `Volume`: Specifies the relative volume to be used when playing the sound. It accepts a value between 0.0 and 1.0, default being 1.0.
- `Panning`: Panning control. -1.0 for full left & 1.0 for full right. Defaults to 0.0 (center).
- `Mix`: Defines the Panning mode. True for mixing channels based on their position and false to act like a regular balance; Defaults to false.
- `DistanceList`: Minimum (mandatory) and maximum (optional) distances for sound spatialization, in units. If any of them is negative, sound spatialization will be disabled for this sound. Defaults to -1.0 # FLT_MAX.
- `GainList`: Minimum (mandatory) and maximum (optional) gain, used to clamp computed gain during sound spatialization, if enabled with DistanceList. Should be within [0.0f - 1.0f]. Defaults to 0.0 # 1.0.
- `RollOff`: RollOff factor used to compute sound spatialization, if enabled with DistanceList. Defaults to 1.0.
- `FilterList`: Filters will be chained in their declaration order.
- `KeepInCache`: Specifies if the sound data (ie. the sample itself) should be kept in memory even when no sound of this type is used. This gives better performances (no disk access) but costs memory. Its default value is `false`.

NB: Spatialized sounds can be performed on either mono or stereo sounds. There is no longer any restriction on sample types.

BiquadFilter

- `UseCustomParam`: When set to true, an event will be sent to override params values at runtime. Defaults to false, ie. no runtime override.

LowPassFilter

- `UseCustomParam`: When set to true, an event will be sent to override params values at runtime. Defaults to false, ie. no runtime override.
- `Frequency`: Cutoff frequency.
- `Order`: Should be between 1 and 8. Cannot be overridden at runtime with UseCustomParam. Defaults to 2.

HighPassFilter

- `UseCustomParam`: When set to true, an event will be sent to override params values at runtime. Defaults to false, ie. no runtime override.
- `Frequency`: Cutoff frequency.
- `Order`: Should be between 1 and 8. Cannot be overridden at runtime with UseCustomParam. Defaults to 2.

BandPassFilter

- `UseCustomParam`: When set to true, an event will be sent to override params values at runtime. Defaults to false, ie. no runtime override.
- `Frequency`: Cutoff frequency.
- `Order`: Should be between 2 and 8. Cannot be overridden at runtime with UseCustomParam. Defaults to 2.

LowShelfFilter

- `UseCustomParam`: When set to true, an event will be sent to override params values at runtime. Defaults to false, ie. no runtime override.
- `Frequency`: Center frequency.
- `Q`: Defines the filter's slope in]0.0-1.0], the higher, the steeper. Defaults to ~0.707107 (1/sqrt(2.0)).
- `Gain`: In dB. Positive for amplification, negative for damping. Defaults to 0 (no change).

HighShelfFilter

- `UseCustomParam`: When set to true, an event will be sent to override params values at runtime. Defaults to false, ie. no runtime override.
- `Frequency`: Center frequency.
- `Q`: Defines the filter's slope in]0.0-1.0], the higher, the steeper. Defaults to ~0.707107 (1/sqrt(2.0)).

- `Gain`: In dB. Positive for amplification, negative for damping. Defaults to 0 (no change).

NotchFilter

- `UseCustomParam`: When set to true, an event will be sent to override params values at runtime. Defaults to false, ie. no runtime override.
- `Frequency`: Center frequency.
- `Q`: Defines the filter's slope in]0.0-1.0], the higher, the steeper. Defaults to ~0.707107 (1/sqrt(2.0)).

PeakingFilter

- `UseCustomParam`: When set to true, an event will be sent to override params values at runtime. Defaults to false, ie. no runtime override.
- `Frequency`: Center frequency.
- `Q`: Defines the filter's slope in]0.0-1.0], the higher, the steeper. Defaults to ~0.707107 (1/sqrt(2.0)).
- `Gain`: In dB. Positive for amplification, negative for damping. Defaults to 0 (no change).

DelayFilter

- `UseCustomParam`: When set to true, an event will be sent to override params values at runtime. Defaults to false, ie. no runtime override.
- `Delay`: In seconds. Cannot be overridden at runtime with UseCustomParam. Should be > 0.0.
- `Decay`: Amount of the “repeat” part, in [0.0-1.0]. 0.0 to deactivate the echo entirely and only introduce delay. Defaults to 0.0.

Latest config settings for the Development Version

We endeavor to keep the config properties on this page up to date as often as possible. For up to the minute config information for the latest version of Orx, check the most recent published at:

CreationTemplate.ini and

SettingsTemplate.ini

Additionally these files can be found under your orx source tree in the `orx/code/bin` folder.

From:
https://www.orx-project.org/wiki/ - **Orx Learning**

Permanent link:
https://www.orx-project.org/wiki/en/orx/config/settings_structure/orxsound

Last update: **2025/09/30 17:26 (14 months ago)**