

Unreal Audio Subsystem

Tim Sweeney

Epic MegaGames[?], Inc.

<http://www.epicgames.com/>

Audience: Level Designers, UnrealScript[?] Programmers, C++ Programmers.

Last Updated: 07/21/99

Audio-related classes

UAudioSubsystem is the abstract base class of the sound subsystem.

The actual sound subsystem is implemented in a subclass of UAudioSubsystem[?]. The reference implementation is UGalaxyAudioSubsystem[?], which supports the Galaxy music and sound system. However, UAudioSubsystem[?] is general enough that it can be ported to other sound systems fairly easily.

USound contains a sound sample with the following characteristics:

- .WAV format.
- 8-bit or 16-bit format.
- Monaural.
- Either one-shot or looped.

UMusic contains a digital music file with the following characteristics:

- 32 or fewer channels maximum (16 or fewer recommended).
- .MOD, .STM, .S3M, .IT format.

UnrealScript[?] interface

PlaySound (Actor class): plays a simple, non-looping sound effect emanating from the actor. Use this for all normal game sound effects. You can specify the following parameters:

- Sound: A sound effect; should be a one-shot (non-looping) sound.
- Slot: One of the eight "slots" to play the sound in. This enables you to prevent more than one sound from playing on a particular slot. For example, when an enemy is shot several times in rapid succession, you want the successive sound effects to cut each other off, rather than hearing a chorus of simultaneous, overlapping screams.
- Volume: Sound volume multiplier. 1.0 means "normal volume", 0.5 means "half volume", 2.0 means "double volume", etc.
- bNoOverride: Whether this sound effect should override another sound playing on the same slot, or be ignored.
- Radius: Radius of the sound in world units, defaults to 4096.0.
- Pitch: Pitch multiplier. 1.0 means "normal pitch", 0.5 means "half pitch".

ClientHearSound (PlayerPawn[?] class): called when a particular PlayerPawn[?] actor hears a sound effect. This is how sounds are sent to clients in a network game. You may call this function directly (in a particular PlayerPawn[?]) to have only that one player hear a sound.

ClientSetMusic (PlayerPawn[?] class): Sets the music for a particular player.

UnrealEd[?] music tips

In UnrealEd[?]'s music and sound browsers, you can listen to music and sound effects by double-clicking on them, or with the Play/Stop buttons. You can hear the level's ambient sounds while editing by clicking on the realtime button in the 3d view (it looks like a joystick icon).

Any actor in Unreal can emit an ambient sound. The properties to check out are AmbientSound[?], SoundRadius[?], SoundVolume[?], and SoundPitch[?]. When you turn on UnrealEd[?]'s "Show Actor Radii" option in a 2d view, you can see the ambient sound radii as blue circles.

In Options/Level, you can set the default music, music section, and cd track for the level.

MusicEvent actors work similarly to SpecialEvent[?] actors, in that they let you cause music changes in response to any event, such as the player touching a trigger, killing a monster, etc. In multiplayer games, MusicEvent[?] actors only cause a music change in the player which

UDN

Search public documentation:

Search

AudioSubsystem

Licensees can [log in](#).

[Red](#) links require licensee log in.

Interested in the Unreal Engine?

Visit the [Unreal Technology](#) site.

Looking for jobs and company info?

Check out the [Epic games](#) site.

Questions about support via UDN?

Contact the [UDN Staff](#)

caused the event, not all players in the game. MusicEvents[?] have a "Song" variable which you can set to "None" if you want to use the song you set for the level in Options/Level.

3D positioning

A sound is positioned in 3d space, positioned at the Location of the actor who played the sound effect or ambient sound.

As the actor that played the sound moves around, the actor's sound is repositioned. This affects the following properties:

- Distance attenuation.
- Doppler shifting.
- Stereo panning (if UseStereo[?] is enabled).
- Dolby Surround Sound positioning (if UseSurround[?] is enabled).

Doppler shifting is based on the emitting actor's Velocity. Note that it is possible for actors to move around directly with SetLocation[?] or MoveActor[?] without having a valid Velocity, thereby causing no doppler shifting.

Dolby SurroundSound[?] software encoding requires a Dolby decoder and four speakers, but it works on all stereo sound cards. It simply pans sound effects and ambient environment sounds between the four Dolby channels (center, rear, left, and right).

Sound slots

Each actor in the world has eight "sound slots", so it can play up to 8 simultaneous (overlapping) sound effects.

When an actor plays a sound in a slot where an old sound is already playing, the old sound can either be cut off and overridden, or the new sound can be ignored (see the bNoOverride parameter in PlaySound[?]).

Sound slot zero (SLOT_None) is special: it indicates that a sound should not interrupt **any other sounds**. For sound effects like rapid-fire gunshots, use SLOT_None to allow multiple instances of the sound to play simultaneously.

In Unreal, there is a hard limit to the number of sound effects which may be **playing simultaneously**. This limit is set in the Unreal.ini file, and can vary from 16 to 32. During gameplay, the sound engine prioritizes each sound effect according to its volume and its distance from the listener, and culls the least important sounds. High-volume sounds override low-volume sounds, and sounds near the listener override sounds far away from the listener.

Sounds

Sound effects can be either 8-bit or 16-bit, and can be sampled at any reasonable rate (4kHz-44kHz). What rate and bit depth to use is entirely a quality vs. memory usage tradeoff. In general, use the lowest bit depth and sampling rate which gives you acceptable sound quality.

For best sound quality, make sure your sound effects are not clipped, i.e. sampled at so high a volume that the tops or bottoms of their waveforms are truncated.

Ambient sounds

Each actor in the level can have one ambient sound effect, which is specified in its AmbientSound[?] property. This should refer to a sound effect which loops. If a non-looping sound effect is used as an AmbientSound[?], it will not play properly (it will only play once).

Persistence and correctness

Sounds have a radius. They diminish linearly according to distance and are culled when too far away to hear. Culling distance is a linear function of the sound's volume.

Ambient sounds persist across loads and saves.

Regular sound effects do not persist across loads and saves. A regular sound effect can only be heard by players (local and network) who can hear it when it begins playing. For example, if you have a 10-second sound effect and you start playing it in a room, then five seconds later a far-away player teleports into the room 5 seconds later, he will not hear the sound at all.

Looping sound effects

To create a looping sound effect in a .wav file, you need a sound editing program like SoundForge[?] (<http://www.sfoundry.com>) which supports looping sound effects. In Sound Forge, follow these instructions:

1. Run SoundForge[?].

2. Load your .wav file (either 8-bit or 16-bit).
3. Bring up the "Special/Edit Sample..." dialog.
4. Click on the "Sustaining" choice (as opposed to "One shot").
5. Select "File/Save As...".
6. Make sure that the "Save Sampler Information in file" option is checked.
7. Click on the "Save button".

Limitations

Software-mixed sound:

- 3D sounds are only stereo panned, but not fully spatialized (i.e. phase shifted and bandpass filtered based on sound absorption in the ear as a function of position).
- The sound system is limited to a maximum of 64 channels of music and sound. By default, 16 sound effects channels are allocated, leaving 48 for music. You can change this from the Unreal.ini file.
- Each playing sound effect, ambient sound, and music channel uses CPU time. Therefore, we recommend keeping the number of sound effects audible to the player fairly low, and keeping music channel usage "lean and mean".

Hardware-mixed sound:

- **3D sounds are spatialized by whatever the hardware supports.** For example, Aureal pans it, phase shifts it, and applies a complex convolution and bandcap filter to model positional modulation of the sound by the head and ears.
- The sound system is limited to X channels of hardware mixed sound (where X depends on the kind of hardware, and can be from 6 to 64 for existing sound hardware). Any sounds for which hardware acceleration is unavailable are emulated in software using the software mixer, and thus may sound different (less spacey) than the same sound played on hardware.

Dynamic music

Each level has a default song and CD track which is played throughout the level. You can specify the song in UnrealEd² in "Options/Level" in the "Audio" tab.

Music can be transitioned dynamically via a SpecialEvent² object hooked up to a trigger or any other event source.

Music is not zone based, because that would become annoying when backtracking through maps and in network play.

Music is tracked per player so it can be used equally well in single player and network play.

Zone-based echo and reverb

The audio subsystem includes a six-tap sound reverberation filter which enables you to customize the echo and reverb effects on a per-zone basis (there can be up to 64 zones in a level, so you can have quite a bit of variety in the audio postprocessing within a **level**).

When you add a ZoneInfo² actor into a zone, you can set the following properties:

- **bool bReverbZone:** True to enable reverberation effects in the zone; False to disable them.
- [this is outdated, but the quick version is: the other properties are 6 reverb tap delays and gains]

You can experiment with reverberation filters quickly by going to Options/Level, setting the reverb effects for the entire level, then double-clicking on sound effect names in the sound browser.

Music formats

Unreal supports two major kinds of music:

- **CD audio.** You can compose your game's sound track, record it on CD, and specify in-level music by CD track. Advantages of CD audio include highest possible music quality, no CPU overhead, and giving composers the freedom to create music however they want. The major drawback is that you can't do dynamic music very well since switching between CD tracks isn't instantaneous, and you can't distribute it via the Internet.
- **Digitally mixed music** (.mod, .s3m, .it, etc). These are special music formats which include digitally recorded instruments as well as sequencing data. Advantages: Easy to distribute on the Internet, and more amicable to dynamic music. Disadvantages: somewhat CPU intensive to mix (3%-5% of CPU time), lower sound quality than CD.

Windows platform implementation notes

We only support 16-bit sound output. We tested 8-bit sound output and the quality was unacceptable low: when you mix 64 channels of sound into an 8-bit output stream, you effectively get 2-bit sound effects. 2-bit sound effects are Not Good.

The sound system supports both the legacy WinMM² sound support, and DirectSound². DirectSound² generally achieves the lowest

latency, while WinMM² works on Win95 machines which don't have DirectSound², and WinNT².

The sound code makes heavy use of MMX to improve mixing quality and speed. You get about a 2X performance and quality boost on Pentium processors with MMX technology.

On non-MMX machines, the sound code makes some quality/speed tradeoffs by limiting sound effects to having only 64 volume levels. You can hear this limitation by setting up an ambient sound effect with a high radius in an otherwise quiet area: the discrete steps between volume levels are quite audible.



Copyright © 2001-2010 [Epic Games, Inc.](#)

[Terms and Conditions](#)