
Subject: Re: Sound device access system based on PortAudio

Posted by [dolik.rce](#) on Thu, 25 Nov 2010 20:25:48 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi everyone,

Good news, we are getting close to something potentially useful here. I added basic support for .wav files and proof of concept function PlayWav(const char* filename). There is also an example how to read informations about the file. The wave file parsing is taken from Timothy J. Weber website. It is written in a code style amazingly similar to U++ and has IMHO quite permissive license (at least it is the closest thing to license I could find).

:Quote:PERMISSION

Copyright is retained by Timothy J. Weber. License is granted to use this source code for any purpose. That should be OK, for all possible uses within U++, right? The only sub-optimal thing about the code is that it uses STL, namely string, stack and vector. It probably can be converted to U++ later...

Now, before I'll continue adding more formats and standardizing the interface for playing audio files, I would like to discuss it here with you. My idea is to have a system similar to image formats - self registering plugins, each handling one audio format (or one family of formats). The question is how should the interfaces look.

The few ideas I have so far: Each format should support simple PlayXYZ(const char*) function, preferably non-blocking.

There should be a Play(const char*) function, that would recognize file type and call appropriate PlayXYZ().

Each format should provide an interface for reading samples from file and also for storing samples to file (decoding/encoding). I'm not sure if they should be separate (e.g. WavEncoder, WavDecoder) or joint (e.g. WavAudio) providing both.

All formats should be accessible in uniform way, that means probably common base class.

The audio files are by nature rather big, so it might be a good idea to provide streaming capabilities (maybe even based on Stream class?) both for reading and writing.

What are your opinions, ideas, recommendations or hints? Any inputs are more than welcomed.

Furthermore, I would like to ask if there is some simple way to make a function non-blocking without requiring MT flag? The only way I can see is using the platform specific threading. Using callbacks is probably not an option here, as the audio playback requires quite high performance...

I'm looking forward to your feedback!

Honza

PS: The PlayWav is just a proof of concept, so don't expect much. It blocks, until the sound is played, which is pretty bad behavior (the GUI doesn't respond). Also, it is quite easy to break and can't play everything (no support for compressed wavs or resampling if the content doesn't match your hardware capabilities)... But it is a start

File Attachments

1) [sound.zip](#), downloaded 356 times
