Subject: Sound device access system based on PortAudio
Posted by dolik.rce on Thu, 04 Nov 2010 20:54:16 GMT
View Forum Message <> Reply to Message

Hi everyone!

Some time ago, I set myself a goal to bring support for sound into U++. Today, I would like to present you first preview of my work in this direction. I chose PortAudio library, which supports great selection of supported platforms and backends (see table below) and the MIT license should be compatible with BSD. This library provides only access to sound devices. to get something really usable, it will have to be extended with some other code that would handle reading, writing and processing of sound files in various formats.

The attached zip archive contains three packages. The biggest one, plugin/portaudio contains sources of Portaudio + few files that take care about proper integration into the u++ build system. Next package, called simply Sound provides U++ wrapper classes that encapsulate the C code from the library. And finally the last package, SoundExample is a simple testing app that gives you some info about available sound APIs, allows to run simple sine wave generator on various sound devices with various options and also contains a simple recording test.

How to compile this: plugin/portaudio can be configured to use various backend via flags (ALSA, JACK, OSS, ASIHPI, DSOUND, WMME, WASAPI, WDMKS). If you don't supply any flags, default one apply (OSS+ALSA on linux, WMME+WASAPI on win32). You can specify multiple flags at once. I recomend using the dot syntax, e.g. "GUI .ALSA .OSS" in order to prevent unnecessary recompilation of everything. For some of the backends to compile/work it might be necessary to have available some libraries or other dependencies (also listed in the table below).

Usual disclaimer applies: This is alpha code, might not be stable and/or might not work as expected. The api in Sound is definitely not stable yet, it will change soon and often. There is no documentation for now, but I promise to write it as soon as the API stabilizes.

Here is a list of supported backends and some info. If you feel like testing on some platforms I didn't have access to (and there is a majority of those), feel free to do so and let me know so I can update the table.

API	Platform	State	Tested	Requirements
jack	unix	compiles	not tested	libjack
oss	unix	works	Arch Linux	none (?)
alsa	unix	works	Arch Linux	libalsa
asihpi	unix	implemented	not tested	???
asio	win&mac	not implemented	---	ASIO SDK, ASIO Drivers, registration
coreaudio	mac	not supported	---	---
dsound	win	implemented	not tested	DirectX SDK
wmme	win	works	wine 1.3.4	none (?)
wasapi	win	probably works	wine 1.3.4	Vista SDK (only for MSVC)
wdmks	win	implemented	not tested	DirectX SDK

More info about the backends (best links I could dig up...):
jack: http://jackaudio.org/
oss: http://www.opensound.com/oss.html
alsa: http://www.alsa-project.org/
asihpi: http://www.audioscience.com/internet/download/linux_drivers. htm
asio: http://www.steinberg.net/en/company/developer.html
coreaudio: http://developer.apple.com/technologies/mac/audio-and-video. html
dsound: http://en.wikipedia.org/wiki/DirectSound
wmme: http://en.wikipedia.org/wiki/Windows_legacy_audio_components
#Multimedia_Extensions_.28MME.29
wasapi: http://msdn.microsoft.com/en-us/library/dd371455(VS.85).aspx
wdmks: http://www.portaudio.com/trac/browser/portaudio/trunk/src/ho stapi/wdmks/readme.txt

I'm looking forward to your comments

Best regards,
Honza

EDIT: Attachment removed in favor of newer version below.