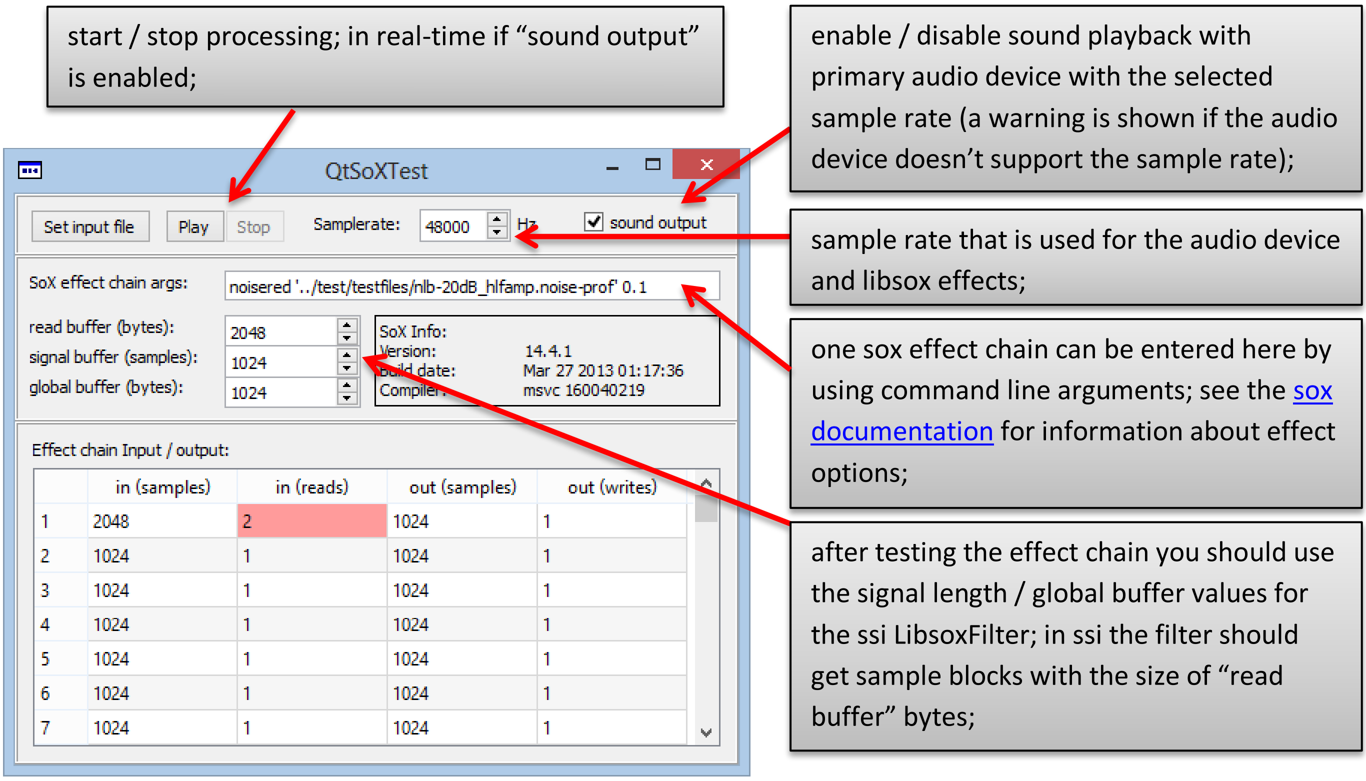
Usage of the QtSoXTest program



start / stop processing; in real-time if “sound output” is enabled;

enable / disable sound playback with primary audio device with the selected sample rate (a warning is shown if the audio device doesn’t support the sample rate);

sample rate that is used for the audio device and libsox effects;

one sox effect chain can be entered here by using command line arguments; see the sox documentation for information about effect options;

after testing the effect chain you should use the signal length / global buffer values for the ssi LibsoxFilter; in ssi the filter should get sample blocks with the size of “read buffer” bytes;

QtSoXTest

Samplerate: 48000 Hz

☑ sound output

SoX effect chain args: noisered '../test/testfiles/nlb-20dB_hlfamp.noise-prof' 0.1

read buffer (bytes): 2048

signal buffer (samples): 1024

global buffer (bytes): 1024

SoX Info:
Version: 14.4.1
Build date: Mar 27 2013 01:17:36
Compiler: msvc 160040219

Effect chain Input / output:

	in (samples)	in (reads)	out (samples)	out (writes)
1	2048	2	1024	1
2	1024	1	1024	1
3	1024	1	1024	1
4	1024	1	1024	1
5	1024	1	1024	1
6	1024	1	1024	1
7	1024	1	1024	1

This program can be used to determine if a SoX effect chain can be used with the specified settings by the ssi plugin “LibsoxFilter”. The program can only read 16 bit integer mono wav files. The ssi plugin also supports 32 bit float and multiple channels. All buffer settings and the “ssi_sample_block_offset” setting should be multiplied by the used channel count for ssi. If memory errors occur in *libsox* with the set buffer sizes this program crashes!

At the beginning you can select a file by clicking the “Set input file” button. After that you can set the different settings and press the “Play” button. If there are errors in the given “SoX effect chain args” the errors are displayed. If you enabled “sound output” the file will be played in real-time – if you disabled it the audio data is processed as fast as possible. If using the audio output mode you don’t have to play the whole file to analyze the effect chain. Just press “Stop” whenever you want. For big wav files you should only use the real-time mode and press “Stop” after some time since it can take quite a while to display many data in the table. The read / write counts of the effect chain are shown in the table. Red marked cells indicate multiple reads / writes that are not allowed in ssi since this isn’t real-time.

SSI filters have to output one sample block for each input sample block immediately. Some *SoX* effects like “noisered” which uses window overlapping with 1024 samples, have to read more than one block at the beginning. If after that a constant write / read count of one sample block is used by the effect chain (see screenshot) the settings are usable by the “LibsoxFilter”. It is possible that you have to try several settings to reach this since some effects produce variable latencies depending on the buffer sizes, sample rate or effect options. Eventually it is not a bad idea to have a look at the code of a specific *SoX* effect or to know its basic design to estimate its latency.

For example if an effect uses fft it has to gather at least as many samples as fft bins are used. “Noisered” uses 2048 fft bins with 50% (1024 samples) window overlapping. In the beginning it has to read at least 2048 samples to be able to call the fft the first time. After that the first 1024 samples are immediately output. For the next outputs it only needs to constantly read 1024 samples to output 1024 samples.

If the test program shows multiple reads at the beginning but constantly uses one write / read after that - like in the screenshot - you can set the “LibsoxFilter” option “ssi_sample_block_offset” to “in (reads)” minus one blocks. The ssi filter will output this count of blocks with zero value samples but also fills the *libsox* pipeline with the input samples. After this block count the effect chain should be saturated and output its own values.

In the table of the screenshot you can see that there were two reads at the beginning (marked red) resulting in one write. After that each read results in one write. So you have to set “ssi_sample_block_offset” to 1. “sox_signal_length” and “sox_global_buffer” should be set to 1024. The “LibsoxFilter” should get “read buffer” bytes as input data from ssi. For example if the data is directly from a ssi “WavReader” the “blockInSamples” should be set to 1024 samples (= 2048 bytes) if using 16 bit integer mono samples.

Resulting option file:

```
<?xml version="1.0" ?>
<options>

    <item name="sox_global_buffer" type="UINT" num="1" value="1024"
help='SoX doc "sox_globals.bufsiz": Default size (in bytes) used by libSoX
for blocks of sample data. Plugins should use similarly-sized buffers to
get best performance.' />

    <item name="sox_signal_length" type="UINT" num="1" value="1024"
help="in / out signal length (in samples) between SoX effects" />

    <item name="sox_effect_chain_str" type="CHAR" num="4096"
value="noisered 'nlb-20dB_hlfamp.noise-prof' 0.1" help="SoX effect chain
arguments" />

    <item name="ssi_sample_block_offset" type="UINT" num="1" value="1"
help="the offset of sample blocks from ssi after which 1 input block
results always in 1 output block" />

</options>
```