The Complete Moresampler Tutorial

April 7, 2016 Kanru Hua UTAU/Moresampler 10 Comments moresampler tutorial

Since the very initial release, numerous features have been integrated into Moresampler, but the old tutorial somehow still targets at the now-deprecated 0.2.0 version. To help you take full advantage of Moresampler, here I've written a much more comprehensive tutorial, or a reference, if you'd like to call it that.

The current version of this tutorial is written for Moresampler 0.7.1. Some features may not work on previous versions.

Contents

1. Overview

Moresampler is a synthesis backend (refered to as "resampler" and "wavtool" in UTAU terminology) for singing synthesis software UTAU. Moresampler has two fundamental differences from most other resamplers. First, it's both a *resampler* and a *wavtool* (and since version 0.7.0, it's also a tool for automatic voicebank configuration from sound files). Second, it is parametric.

What "parametric" means is: Moresampler first analyzes a speech sample, converting it from waveform to Moresampler's own data format, `.llsm` file. Then it loads parameters from `.llsm` files, modifies them according to UTAU's command, and finally renders the modified parameters back to waveform.

When Moresampler is launched as wavtool, it loads temporary `.llsm` files generated by itself in resampler mode, instead of reading `.wav` files. These `.llsm` files are parameters for generating each segment of singing. Moresampler joins these parameter segments into a whole song, and synthesizes them into waveform after reaching the last note.

The major advantages of being parametric are

1. Moresampler is free from wave interference, which distorts the voice if waveform segments are simply cross-faded together.
2. Vocal effects are easily implemented in LLSM parameter domain. LLSM stands for Low Level Speech Model. It is based on latest research outcomes in the field of speech synthesis.

Like many other resamplers, Moresampler also has its own frequency table file, which is `desc.mrq` file, introduced since version 0.3.0. However, `desc.mrq` file is accessed only once when Moresampler generates `.llsm` files, since `.llsm` file already completely describes the speech sample.

RECENT POSTS

libllsm2 Release

Help me with revising the experimental Arpasing recording script

This blog might remain broken for an indefinite amount of time.

Announcing SHIRO the Speech Alignment Toolkit

Moresampler 0.8.3 Release

CATEGORIES

Acoustics/Electrical

Corpus Design

Speech Processing

Uncategorized

UTAU/Moresampler

ARCHIVES

December 2017

August 2017

July 2017

June 2017

May 2017

March 2017

February 2017

October 2016

August 2016

July 2016

June 2016

May 2016

April 2016

March 2016

February 2016

RECENT COMMENTS

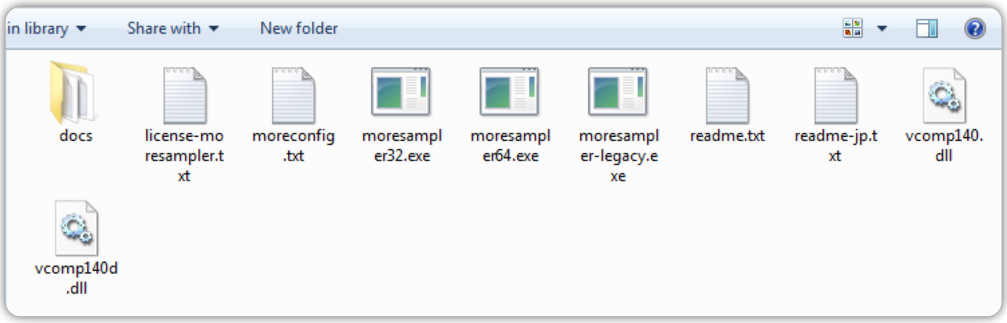
Design Philosophy

The design philosophy behind Moresampler is that, without any flags, Moresampler should produce the most neutral voice that acoustically best matches the quality of the input samples. However being neutral doesn't always imply that it is the best expected voice quality. That's why Moresampler has a wide range of flags for shaping the voice quality.

2. Setting up and basic usage

The latest release can be downloaded here. Versions prior to 0.3.0 are here, though they are not compatible with newer versions.

It's recommended to keep all files in the same directory as they were extracted from the zip archive. `moreconfig.txt`, `vcomp140.dll` and `vcomp140d.dll` need to be in the same directory with the executable. The `.dll` files are only required for versions prior to 0.7.0.

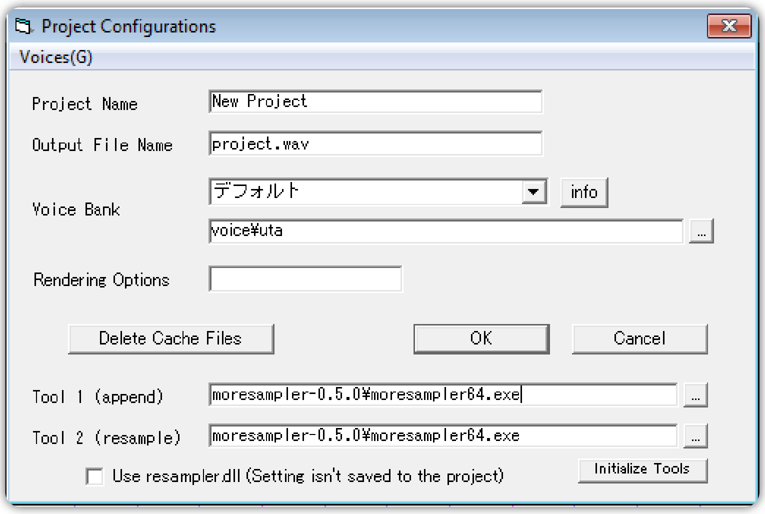


Since version 0.5.0, there are three executables in each release, namely `moresampler32.exe`, `moresampler64.exe` and `moresampler-legacy.exe`. In later versions (>= 0.7.0), only the compatibility version is kept.

`moresampler32.exe`	32-bit version
`moresampler64.exe`	64-bit version
`moresampler-legacy.exe`	32-bit version compatible with old systems (e.g. *Windows XP*) with multi-threading disabled

So normally you would use `moresampler32.exe` or `moresampler64.exe` depending on your system. Note that even if your CPU is 64-bit, the 64-bit version still won't work if you are running a 32-bit operating system. In case Moresampler doesn't work, you should try running moresampler-legacy.exe, which has the best compatibility.

To start using Moresampler, go to UTAU's project setting panel, select Moresampler in both Tool1 and Tool2 path. If you are using *presamp* as a wrapper, you need to specify Moresampler in the *predit* plugin.



Project Name: New Project
Output File Name: project.wav
Voice Bank: デフォルト
voice¥uta
Rendering Options:
Tool 1 (append): moresampler-0.5.0¥moresampler64.exe
Tool 2 (resample): moresampler-0.5.0¥moresampler64.exe
☐ Use resampler.dll (Setting isn't saved to the project)

Note: when using Moresampler as Tool1, Tool2 can't be resamplers other than Moresampler itself. For more information on compatibility issues, please visit the compatibility section.

3. Configuration file

`moreconfig.txt` is the configuration file for Moresampler. In contrast to flags, configuration affects Moresampler **globally and indiscriminately with respect to notes**. Moresampler supports two types of configuration files: global configuration and voicebank configuration. Global configuration is located

pune on Moresampler 0.8.3 Release

Dela Livi on Moresampler Bug Report Page

Dela Livi on Moresampler Bug Report Page

jiten on Moresampler Bug Report Page

Noida Escort on Moresampler 0.8.3 Release

under Moresampler's directory and is distributed with Moresampler; Voicebank configuration, on the other hand, is located under voicebank's directory and distributed with the voicebank (visit "Voicebank configuration" and "Tips for voicebank authors" sections for how to setup Voicebank configuration).

A configuration file contains multiple, unordered entries of options in key-value pair format. There are four broad categories of entries:

Category	Description
Analysis configuration	Options that are effective when Moresampler generates/regenerates `.llsm` and `desc.mrq` files; they affect the analysis algorithm. Once `.llsm` file is generated, these options would not have any effect
Synthesis configuration	Options that are effective when Moresampler modifies LLSM parameters
Output configuration	Options that control the output format
Miscellaneous configuration	Other options that do not fit in the above categories
User configuration	Options for customizable features (e.g. meta flags)

The syntax of each entry is defined as follows,

```
<#><*>key value
```

where <> means optional token(s). A line starting with `#` is commented and Moresampler will skip this line no matter what's after the `#` character; a line starting with `*` means overwriting voicebank configuration, if there exists any (discussed in detail in "Voicebank configuration" section). The following are examples of syntactically correct entries,

```
output-sampling-rate 44100

dump-log-file C:\aaa bbb\log.txt

# blahblahblah 222 333 444 555

*analysis-f0-min 70.0
```

Note that there shouldn't be any space before `key`, # or `*`, nor any space after `value`. There should be only one space between `key` and `value`. In addition, `value` may contain space(s), like the second line in the above example.

List of Output Configurations

Name	Allowed Values	Description
`output-sampling-rate`	Integer	Sampling frequency of output `.wav` file
`output-bit-depth`	8/16/24/32	Bit depth of output `.wav` file
`resampler-compatibility`	`on`/`off`	When turned on, generate `.wav` file in resampler mode (to be compatible with other wavtools); slows down synthesis.

List of Synthesis Configurations

Name	Allowed Values	Description
`synthesis-utau-style-normalization`	`full`/`voiced`/`off`	Apply an adaptive gain to each note such that the peak of synthesized waveform goes to half of the maximum amplitude when volume is 100%; `full`: gain both voiced and unvoiced parts; `voiced`: only gain voiced part; `off`: do not adjust volume. Example (`full`, `voiced`, `off` from top to bottom).
`synthesis-loudness-preservation`	`on`/`off`	Retain the perceived loudness after modification, based on a psychoacoustic loudness measure. Example (`off`, `on` from top to bottom).
`synthesis-duration-`	`auto`/`stretch`/`loop`	Determines how Moresampler extends the duration of each note; `auto`: automatically

`extension-method`		stretch or loop the note based on its original and target duration. The effect can be overwritten by `e` and `Me` flags.

List of Misc Configurations

Name	Allowed Values	Description
`multithread-synthesis`	`full`/`on`/`off`	When turned on, the final synthesis stage in wavtool mode will run in multiple threads (which means faster). When set to "full", resampler mode will also become multithreaded. "multithread", which packs multiple threads into a process, is inherently different from "multiprocess" which launches multiple instances of Moresampler at a time. Note: this feature is not supported by `moresampler-legacy.exe`.
`auto-update-llsm-mrq`	`on`/`off`	Check the last modified time of `.wav` file and corresponding `.llsm` file and mrq data entry. If the `.wav` file is newer than the `.llsm` file, then reanalyze. If the `.wav` file is also newer than the mrq data entry, then reestimate pitch before reanalyzing `.llsm`. This feature might be helpful for voicebank developers.
`dump-log-file`	file path (e.g. `D:\log.txt`)/`off`	Output debug information into a specified file path.

List of Analysis Configurations

Note that once `.llsm` and `.mrq` files are generated, Moresampler will skip analysis procedure. To put the following options into effect, you need to delete all existing `.llsm` files before running Moresampler. For options relating to pitch (fundamental frequency, or f0) analysis, you should also delete `desc.mrq` files, or use *frqeditor* to re-analyze the samples.

For specific examples on using analysis configurations to fix improperly analyzed samples, please visit "Tips and troubleshooting" section.

Name	Allowed Values	Description
`analysis-f0-range-from-path`	`on`/`off`	Infer the pitch range from directory name. For example, if the given sample is under a directory named "C_D4", Moresampler would run pitch analysis in a range close to D4 (around 294Hz).
`analysis-biased-f0-estimation`	`on`/`off`	(over)Emphasize voicing probability during joint pitch & voicing activity estimation, followed by a pitch & voicing correction procedure; tend to reduce false negative but raise false positive rate; works for noisy/coarse speech but **degrades the quality of clean/smooth speech**.
`analysis-f0-min`	positive real number	The lower bound for pitch (in Hz).
`analysis-f0-max`	real number, greater than `analysis-f0-min`	The upper bound for pitch (in Hz).
`load-frq`	`strict`/`on`/`off`	When `desc.mrq` is not available, load pitch data from `.frq` file when set to `strict`; when set to `on`, use pitch data from `.frq` file to correct the pitch estimated by Moresampler's own estimator. No matter this option is turned on or not, the result will always be written into `desc.mrq`. Note: unless carefully corrected, pitch provided by `.frq` files are often **not** robust enough for Moresampler to run properly.
`analysis-anti-distortion`	`on`/`off`	When turned on, Moresampler will automatically fix analysis inaccuracy caused by noise distortion or low volume (quantization error) which may result in "sharp", "gross" voice after pitch shifting. However

		turning on this feature may (in theory) slightly blur the speech. Example (`off`, `on` from top to bottom).
`analysis-noise-reduction`	`on`/`off`	Automatically reduces noise when analyzing LLSM from `.wav`; works better with longer recordings.
`analysis-suppress-subharmonics`	`on`/`off`	Automatically remove the subharmonics (if there's any) from input speech during analysis. Might be helpful for screamy voices but slightly degrades the quality of breathy voices.

Voicebank configuration

Voicebank configurations are `moreconfig.txt` files distributed with a voicebank, in the same directories where `.wav` files are stored. Introduced since version 0.6.0, this feature allows voicebank authors to tune up Moresampler for their own voicebanks, and users don't have to set up configuration themselves.

To create a voicebank configuration, simply create a `moreconfig.txt` under the directory where `.wav` files are stored. Only **analysis** and **synthesis** configurations are supported in voicebank configuration.

A configuration file only affects the samples placed under the same directory (excluding subdirectories), which means for a multi-pitch or multi-expression voicebank, you need to create one `moreconfig.txt` for each entry. If the same option is also specified in global configuration file, by default the one in global configuration will be overwritten by the one in the voicebank, unless the one in global configuration file has a prefix '`*`'.

For example, if a UTAU installation contains the following list of files:

```
voice/voicebank1/C4/oto.ini
voice/voicebank1/C4/moreconfig.txt
voice/voicebank1/C4/a.wav, voicebank/C4/b.wav, voicebank/C4/c.wav...
voice/voicebank1/G4/...
voice/voicebank1/C5/...
moresampler/moresampler32.exe
moresampler/moreconfig.txt
```

and the content of `voice/voicebank1/C4/moreconfig.txt` is,

```
analysis-noise-reduction on
synthesis-duration-extension-method stretch
analysis-f0-min 200.0
analysis-f0-max 300.0
```

while the content of `moresampler/moreconfig.txt` is,

```
synthesis-duration-extension-method auto
analysis-noise-reduction off
*analysis-f0-min 60.0
analysis-f0-max 800.0
```

then the first, second and fourth line in `voice/voicebank1/C4/moreconfig.txt` will overwrite the second, first and fourth line in `moresampler/moreconfig.txt`, respectively. However, since the third line in global configuration has prefix '`*`', in contrary it overwrites the third line, "`analysis-f0-min 200.0`" in voicebank configuration. As a result, the options applied on `a.wav`, `b.wav` and `c.wav` under `voice/voicebank1/C4/` are

```
synthesis-duration-extension-method stretch
analysis-noise-reduction on
analysis-f0-min 60.0
analysis-f0-max 300.0
```

4. Flags

Moresampler is compatible with most of the standard resampler's flags. In addition, it's equipped with a new set of flags for adjusting timbre and creating all kinds of vocal effects; those are called "Moresampler Extension Flags".

Flags are case sensitive. For example '`Me`' and '`ME`' have different meanings.

This tutorial would **not** give any recommendation on whether or not to use certain flag on certain kind of voicebanks. Instead, examples of using the flag are given, and the decision on using is totally up to you.

Standard UTAU flags

The following is a list of standard resampler's flags supported by Moresampler, also including some of the flags compatible with *tn_fnds* resampler.

Name	Range	Default	Description & Example
`g`	[-100, 100]	0	Alter the perceived gender of the voice. Positive: male; negative: female. Example (original sample followed by resynthesis with flag '`g40`', voicebank: 京歌カオル) 00:00 00:00
`t`	[-1200, 1200]	0	Shift the pitch by certain cents. 1 cent = 1/100 semitone.
`P`	[0, 100]	86	Peak compressor. When set to 100, it normalizes the peak of output waveform to half the maximum level. When set to 0, it doesn't normalize the output at all. For a number between 0 and 100, the degree of normalization is interpolated. This flag is only effective when "`synthesis-utau-style-normalization`" is not `off`.
`A`	[-100, 100]	0	Amplitude modulation. This flags modulates the amplitude in correlation with change of pitch. It could be helpful for creating realistic vibratos. The sign (positive or negative) controls the direction of such modulation. The formula for amplitude gain is $2^{10^{-5}A\frac{d}{dt}c(t)}$, where c(t) is a function mapping time (seconds) to pitch (cents).
`b`	[-20, 100]	0	Amplitude gain for unvoiced consonants. This flag amplifies or attenuates unvoiced consonants (e.g. /t/ /k/ /s/) by a factor of 0.05 times the number after b. It has less or no effect for voiced consonants (e.g. /g/, /m/). Example (original sample followed by resynthesis with flag '`b-15`' and then '`b20`', voicebank: 京歌カオル) 00:00 00:00
`e`	None	None	Force Moresampler to extend sustained vowels by stretching (time scaling), as opposed to looping. A related flag is `Me`, which has the exactly opposite effect (force looping). By default the duration extension method is specified by "`synthesis-duration-extension-method`" option.
`u`	None	None	Send input directly to output without any processing (such as pitch shifting, time scaling or timbre adjustment). This flag is useful for adding sound effects, for example, breathing sound to the project because you may not want them to be pitch-shifted. This flag is equivalent to a less-known feature in UTAU: adding `$direct=true` to the note settings.

Moresampler Extension Flags

Name	Range	Default	Description & Example
`Mt`	[-100, 100]	0	Tenseness - the extent to which the vocal folds are stressed or relaxed. Positive values correspond to tenser voice quality and vice versa. Example (synthesis without flag followed by synthesis with '`Mt50`' and then '`Mt-50`', song: Unravel, UST by slowbuns, voicebank: 闇音レンリ) 00:00 00:00
`Mb`	[-100, 100]	0	Breathiness. Positive values correspond to breathier voice and negative values reduce the breathing noise. When set to 100, the voice completely becomes whispering. Example (synthesis without flag followed by synthesis with '`Mb-50`',

			then '`Mb50`' and '`Mb100`', song: End of Rain, UST by cilla, voicebank: 闇音レンリ) 00:00 00:00
`Mo`	[-100, 100]	0	Openness - the degree of jaw opening during phonation. Positive values correspond to wide opening and vice versa. Example (synthesis without flag followed by synthesis with '`Mo40`' and then '`Mo-40`', song: Reverse+, UST by rokurin, voicebank: 波音リツ Kire) 00:00 00:00
`Mr`	[-100, 100]	0	Resonance. This flag creates a "singer's formant" around 3kHz if set to positive; otherwise it reduces the formant.
`Md`	[-100, 100]	0	Dryness - the degree of amplitude modulation received by breathing noise due to the periodicity of glottal air flow. The effect of this flag is very subtle and mostly takes place in high frequency band (usually above 6kHz). Example (resynthesis without flag followed by resynthesis with '`Md-100`' and then '`Md100`', voicebank:波音リツ) 00:00 00:00
`MC`	[0, 100]	0	Coarseness - add a roar-like noise to the voice. Example (synthesis without flag followed by synthesis with '`MC50`' and then '`MC100`', song: Tokyo Teddy Bear, UST by UtauReizo, voicebank: 欲音ルコ Kire) 00:00 00:00
`MG`	[0, 100]	0	Growl effect - its name is self-explanatory. Example (synthesis without flag followed by synthesis with '`MG50`' and then '`MG100`', song/UST/voicebank: same as above) 00:00 00:00
`MD`	[0, 100]	0	Distortion effect - an effect similar to growl but vibrates faster. Example (synthesis without flag followed by synthesis with '`MD50`' and then '`MD100`', song/UST/voicebank: same as above) 00:00 00:00
`Ms`	[0, 10], Integer	0	Stabilization - fixing the occasional 'pops' that mostly occurs when shifting down the pitch. It is basically a runtime version of "`analysis-anti-distortion`" with adjustable degree of stabilization (higher number corresponds to stronger stabilization). This flag is recommended when the popping only occasionally occurs, otherwise turn on `analysis-anti-distortion` instead. Example (down-shifting pitch without/with flag '`Ms5`', voicebank: スズ - Ⅹ CROSS-) 00:00 00:00
`Mm`	[0, 100]	100	Model interpolation - interpolating between the classical speech model used before version 0.3.0 and the novel model used since then. By default Moresampler uses the new model (`Mm100`).
`ME`	[-100, 100]	0	Formant emphasis - given positive values, it emphasizes the formants; given negative values, the voice becomes fuzzy.
`Me`	None	None	Force looping - the opposite of '`e`' flag.

Meta flags

Hand-specifying flags for each note could be an extremely tedious job, especially when you're constantly switching between several sets of flag combinations. Introduced since version 0.7.0, meta flag offers a shortcut that you can combine multiple flags into one which saves effort when typing flag sequences in UTAU's note settings panel. Meta flags are defined in the global configuration file, and activated in the format `M+number` (e.g. `M1`, `M2`, `M3`). To define a meta flag, add the following into `moreconfig.txt`,

```
meta-flag-1 MG50MD30MC20Mb30Mt50
```

which is an example defining meta flag `M1`, equivalent to the flag sequence `MG50MD30MC20Mb30Mt50`. Similarly '`meta-flag-2`' corresponds to `M2` and so on.

By putting dot and number after a meta flag, the effectiveness can be scaled by the number (as a percentage, as long as the result of scaling is still within the allowed range of each flag). For the meta flag definition in the previous example, flag sequence '`eMo20M1.50`' expands to '`eMo20MG25MD15MC10Mb15Mt25`'.

5. Tips and troubleshooting

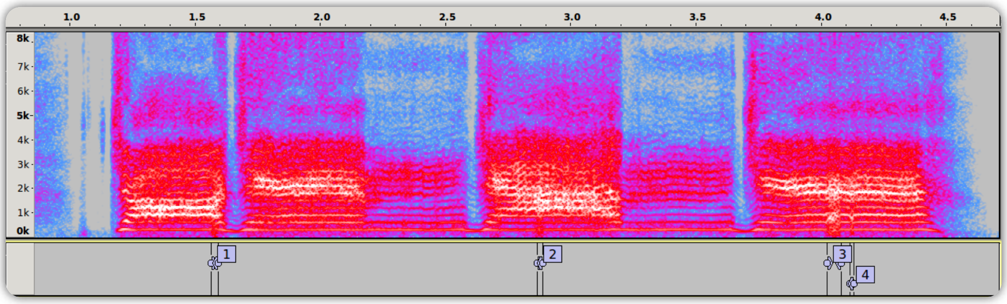
Fixing pops/glitches/noises

These errors (note: not bugs) could result from different reasons. Usually you can identify the cause by looking at the spectrogram of the output `.wav` file, and fix the problem by yourself.

Fixing errors caused by wrong pitch estimation

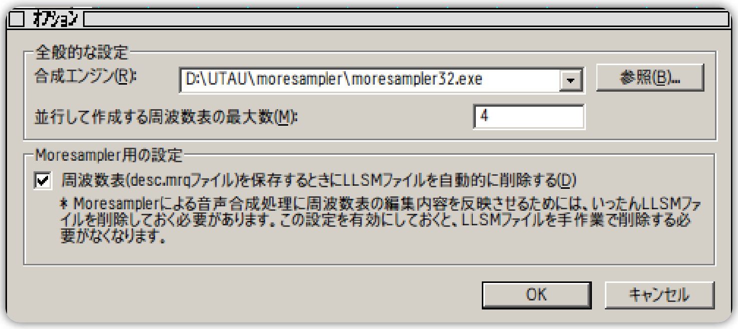
Pitch estimation errors are the most common type of error causing pops and noises. Moresampler's algorithm relies on precise voicing/unvoicing (which means whether the vocal folds are oscillating and pitch exists) detection, and is thus sensitive to voicing and pitch estimation errors. Though by turning on "`load-frq`" Moresampler can also load `.frq` files generated by other resamplers, the voicing estimation provided by these files are often not robust enough for Moresampler to run without pops. That's why by default Moresampler uses its own `.mrq` format and pYIN algorithm, a robust pitch and voicing estimator based on auto-correlation and Hidden Markov Model. pYIN has significantly reduced the error rate. However, it's still hard to completely get rid of errors.

These errors are easy to identify in frequency domain. The following is the spectrogram of a resynthesized sample from a growling voicebank, in which the strong growling effect reduces the periodicity of speech signal and sometimes results in pops, as labelled in the track below. We can see that the harmonics disappear in the labelled ranges and are replaced by some intense noises.



To fix pitch estimation errors, all you need to do is to manually edit `desc.mrq` files. An editor compatible with `.mrq` files is *frqeditor* (version 20160410 or later) written by Mr. Masao. After editing the `desc.mrq` file, save it and delete the corresponding `.llsm` file so Moresampler would generate it again using the manually corrected pitch and voicing information.

Once you open *frqeditor*, set the engine to Moresampler as follows. It's recommended to check the automatically delete `.llsm` file option so you don't have to take efforts to find the file and delete it by hand.



全般的な設定

合成エンジン(R): D:\UTAU\moresampler\moresampler32.exe

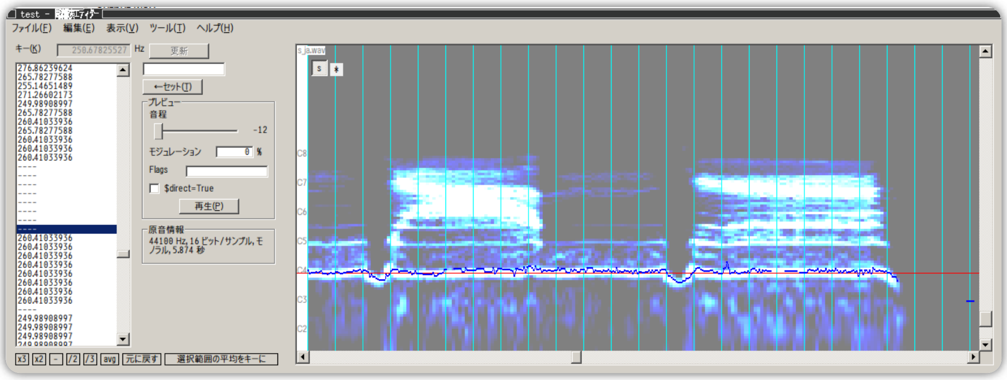
並行して作成する周波数表の最大数(M): 4

Moresampler用の設定

☑ 周波数表(desc.mrqファイル)を保存するときにLLSMファイルを自動的に削除する(D)

＊Moresamplerによる音声合成処理に周波数表の編集内容を反映させるためには、いったんLLSMファイルを削除しておく必要があります。この設定を有効にしておくと、LLSMファイルを手作業で削除する必要がなくなります。

Then open the directory (**not the file itself**) which contains the wrongly analyzed file in *frqeditor*. A list of `.wav` files and the availability of `.mrq` entry will show up in the floating panel. Select the file whose pitch/voicing estimation went wrong and its spectrogram will be loaded into the main window, with a pitch curve overlaid on the top. In the following screenshot it shows that the pitch curve is discontinuous at several positions where harmonics exist and the speech is supposed to be voicing. Connect the discontinuous parts by dragging the mouse with left-button pressed down. To set a region to be unvoiced, just hold right-button and drag the mouse. Please pay attention not to draw pitches over the aperiodic/inharmonic regions.



キー(K) 250.67825527 Hz

```
276.86239624
265.78277588
255.14651489
271.26602173
249.98908997
265.78277588
260.41033936
265.78277588
260.41033936
260.41033936
260.41033936
----
----
----
----
----
----
----
----
260.41033936
260.41033936
260.41033936
260.41033936
260.41033936
260.41033936
260.41033936
260.41033936
----
249.98908997
249.98908997
249.98908997
```

音程 -12

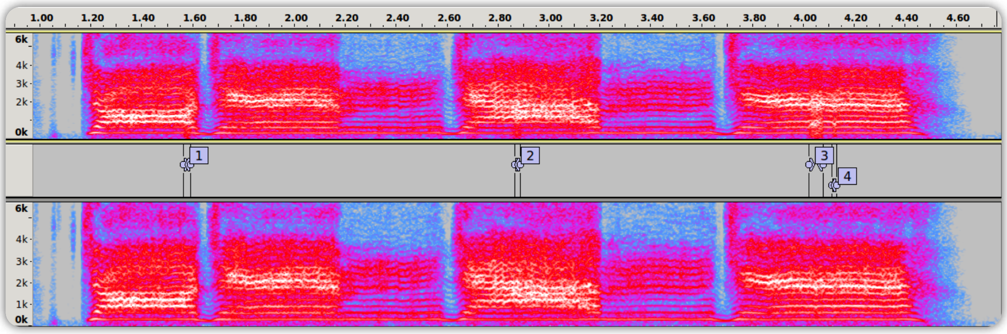
モジュレーション 0 %

Flags

☐ $direct=True

44100 Hz, 16 ビット/サンプル, モノラル, 5.874 秒

Finally, save the `desc.mrq` file and run synthesis again in UTAU. The file will be re-analyzed and the output, in this example, should look like the third track (from top to bottom) in this screenshot:

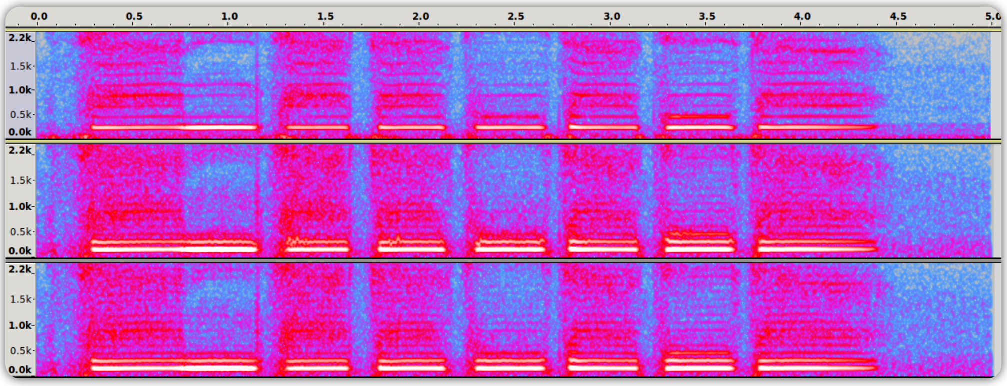


Fixing errors caused by noise

Background/breathing noise could distort the harmonics with low amplitude, which means turning on "`analysis-noise-reduction`" wouldn't completely solve the problem; you also need "`analysis-anti-distortion on`" to correct the periodic component.

The tricky thing is that such distortion is usually hidden when there's no pitch-shifting. The problem could be revealed by shifting (usually down-shifting) the pitch by a few tones, but it could "disappear" if the pitch is shifted by another few tones.

The following is an example of speech distorted by breathing noise. From top to bottom, the first track is the spectrogram of the original sample; the second track is resynthesized version with its pitch down-shifted by an octave; the third track is pitch-shifted version with "`analysis-anti-distortion on`". Notice that the second harmonic in the second track has spurious discontinuities at several positions; the speech in third track has a much smoother second harmonic.



If the noise distortion is systematic (i.e. it appears on lots of `.wav`), you should consider adding "`analysis-anti-distortion on`" to voicebank configuration, or inform the voicebank author to do so. If the distortion is occasional, then just use '`Ms`' flag (e.g. '`Ms5`') instead, which has the same effect but being a runtime version of "`analysis-anti-distortion on`".

Useful tricks

Combination of MEFs

By using several Moresampler Extension Flags together, Moresampler can cover a wide range of voice qualities using limited samples. The flags you may find most useful are `Mt` (tenseness), `Mb` (breathiness), `Mo` (openness) and `ME` (formant emphasis). The exact opposite of a voice effect can usually be achieved by simply inverting the sign.

For example, the combination "`Mt30Mo20Mb-30ME20`" gives a clear voice. The following is a sound sample from End of Rain, feat. 闇音レンリ.

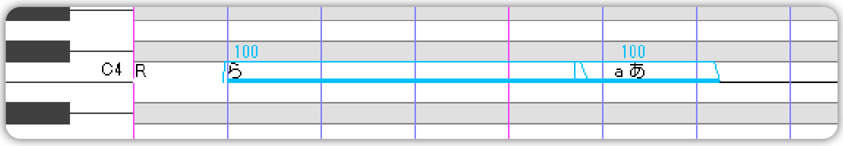
Flags	Synthesis result	
Rendered without any flag	00:00	00:00
Rendered with "`Mt30Mo20Mb-30ME20`"	00:00	00:00
Rendered with "`Mt-30Mo-20Mb30ME-20b-10`"	00:00	00:00

In the last version with inverted flags ("`Mt-30Mo-20Mb30ME-20b-10`"), which give a weak and breathy voice, `b-10` is added to counteract the gain on unvoiced consonants under the effect of "`synthesis-utau-style-normalization full`" (since the volume of vowels has decreased after going through these flags).

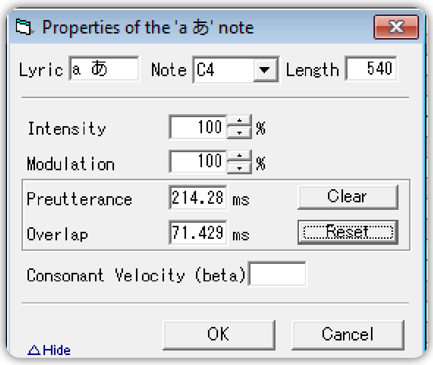
Cross synthesis

Moresampler supports a non-obvious feature which is the ability to crossfade very long notes. While this is possible for other wavtools, so far Moresampler produces the best result since it is immune to interference that could damage the vowel transition. To make best use of this feature, we can create transition between notes of the same vowel but with different flags or voice attributes. (Note: this feature only works when Moresampler is used as both resampler and wavtool.)

As an example, we create two notes (romaji: "ra" and "a a" respectively) having the same vowel "a", with the first note being significantly longer than the second,

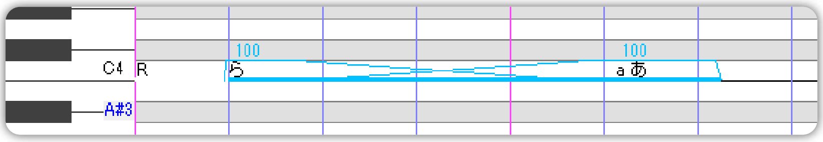


Then go to the properties panel on the second note; click "reset" on Preutterance and Overlap settings,


Properties of the 'a あ' note

Lyric: a あ
Note: C4
Length: 540
Intensity: 100 %
Modulation: 100 %
Preutterance: 214.28 ms
Overlap: 71.429 ms
Consonant Velocity (beta):

We want to increase the duration of transition, so increase both Preutterance and Overlap by the same amount. In this example we add 1500ms to both values, which is slightly shorter than the length of the first note. Press "OK" to apply the changes. Then select both notes and "set crossfade envelopes by p2 and p3". The result would look like:



Now give the second note a "`Mo50`" flag. Render the project and you will hear the voice gradually becoming more "powerful".

Settings	Synthesis result (voicebank:京歌カオル)	
Crossfading without flag	00:00	00:00
Crossfading with no flag and then "`Mo50`"	00:00	00:00

Of course you can use this trick in conjunction with "Combination of MEFs" and/or using samples from different variants of a voicebank (e.g. 波音リツ Kire and 波音リツ Normal).

Bug reporting

Bug reports are welcomed in any language (we use machine translation for languages other than English, Japanese and Traditional/Simplified Chinese). However before sending a bug report, please **make sure you've tried the latest version and the aforementioned tips on troubleshooting** - some errors may not be bugs.

Moresampler comes with a feature that helps us locating the bug by generating a lengthy log file tracking down its own behavior. This feature is enabled by supplying "`dump-log-file`" option with a file path such as "`D:\moresampler-log.txt`". Moresampler may not be able to create the log if the path is under a system-owned directory (e.g. `C:\`).

Please render **only once** in UTAU so the log file won't become messy. Then send the log to the author via email (k.hua.kanru [at] ieee.org). You are also encouraged to comment under the bug report page - doing so would help us keep track of progress and inform other users of which bug has been fixed already.

Your feedback would be more informative if you can attach the output `.wav` (if it didn't crash before generating a wave file) and temporary files created by UTAU. These temporary files are usually stored under `C:\Users\<username>\AppData\Local\Temp\utau1\` where you can find `temp.bat` and `temp.wav`.

Compatibility

Operating systems

Moresampler is developed on *Linux* but for *Windows*. It has been reported to be fully compatible with *Windows 7*, *Windows 8/8.1* and *Windows 10*.

Since version 0.3.1, Moresampler also works on *Linux* through *wine (>= 1.9.2)*. Here's a tutorial on setting up UTAU on *Linux*. However, UTAU seems to encounter some timing inaccuracies on *Linux*.

Moresampler is **possible** to run on *OSX* also through *wine* but we haven't tested yet.

Other resamplers/wavtools

As a resampler (Tool2), Moresampler can be used in conjunction with other wavtools but some options in output configuration has to be changed:

```
output-sampling-rate 44100
output-bit-depth 16
resampler-compatibility on
```

Note: by doing so Moresampler will lose the advantage of minimal interference and arbitrary output sampling rate/quantization level.

As a wavtool (Tool1), Moresampler **cannot** be used with other resamplers because it expects LLSM data files instead of wave files as input.

Moresampler is compatible with *presamp*, when the above conditions are satisfied, i.e. either used as both Tool1 and Tool2, or used as Tool2 only with `resampler-compatibility on`.

Moresampler is compatible with *utaugrowl* only when it's used with other wavtools. Basically there would be compatibility issues when Moresampler is used with plugins that try to access/modify wave files generated by resampler (Tool2). The reason has been explained in section Overview.

Compatibility across different versions of Moresampler/`.llsm` files

It's **not** recommended to mix different versions of Moresampler for Tool1 and Tool2. The author does not guarantee proper functioning when Tool1 and Tool2 are set to different versions of Moresampler.

`.llsm` file format has been constantly updated along with Moresampler. In some cases, Moresampler is backward compatible with `.llsm` files generated by a previous version, and (partially) forward compatible with `.llsm` files generated by a newer version. Occasionally a new release will be incompatible with all previous version. This is done on purpose, usually when we have made certain improvement (or improvement accumulated over several minor versions) on analysis algorithm, so that Moresampler will be forced to update the data records to the latest version. For example, Moresampler 0.6.1 uses the same `.llsm` format as 0.6.0, but it rejects files generated by versions prior to 0.6.0.

- When Moresampler detects a `.llsm` file with **older but compatible** version, it would load the file but also give a warning suggesting you deleting the `.llsm` files;
- when it detects a `.llsm` file with **older and incompatible** version, it regenerates and overwrites the file;

- when it detects a `.llsm` file with **newer but compatible** version, it would load the file but also give a warning suggesting you updating Moresampler;
- when it detects a `.llsm` file with **newer and incompatible** version, it halts and gives an error so as to prevent itself from downgrading the file.

Here is the compatibility matrix across Moresampler and `.llsm` files of different versions,

		Moresampler version		
		< 0.7.0	0.7.0	0.7.1
`.llsm`	< 0.7.0	see table below		
version	0.7.0		=	-
	0.7.1		+	=

							Moresampler version (pric	
		0.1.5	0.2.0	0.2.1	0.2.2	0.2.3	0.3.0	0.3.1
`.llsm`	0.1.5	=						
version	0.2.0		=					
	0.2.1		+	=				
	0.2.2		+	+	=	-		
	0.2.3		+	+	+	=		
	0.3.0						=	-
	0.3.1						+	=
	0.5.0							
	0.6.0							
	0.6.1							
	0.6.2							
	0.6.3							
	0.6.4							

(= indicates full compatibility, +/- indicates forward/backward compatibility)

6. Tips for voicebank authors

There are certain files voicebank authors can distribute (by packing the files into their voicebanks) to improve Moresampler's performance on their voicebanks. In particular `desc.mrq` and `moreconfig.txt` (as voicebank configuration).

However we **do not** recommend distributing `.llsm` files with the voicebank because

- Newer versions of Moresampler won't update these `.llsm` files if they are backward compatible (see section on compatibility). For example, if `.llsm` files generated by Moresampler 0.5.0 are included in a voicebank, then Moresampler 0.6.0 won't automatically update them;
- In the other case, if the newer Moresampler is not compatible with the old `.llsm` files, these files will be replaced by the new version. The old `.llsm` files would be completely ignored;
- Users have to delete these files if they have customized global analysis configurations;
- They increase the total size of voicebank by around 70%.

The general way to quickly tune up Moresampler for your voicebank is to first set up `moreconfig.txt` under **all directories that contain `.wav` files**, then use *frqeditor* to generate `desc.mrq` files and manually correct pitch estimation errors. The first step, setting up `moreconfig.txt` aims at eliminating most pitch estimation errors, though there could still be a few left. Then the second step is meant to correct the rest errors by hand.

If your voicebank has multiple pitches each organized in a directory whose name is the pitch (e.g. "D4", "a#3"), regardless of being upper or lowercase, or some text followed by an underline and the pitch (e.g. "abc_D4", "X_C4"), then all you need to do is to simply let Moresampler extract the pitch range from the file path. Your voicebank configuration under all directories should have this line: `analysis-f0-range-from-path on`

Otherwise, you need to supply Moresampler with a frequency range for each directory. Here is a nice webpage for translating pitch to frequency in Hertz. Typically the frequency range for a set of samples at fundamental frequency x is from 0.6x to 1.7x. So if all samples under a directory are at C4, the frequency range would be from 0.6 * 262 = 157.2Hz to 1.7 * 262 = 445.4Hz and `moreconfig.txt` under this directory should contain

```
analysis-f0-range-from-path off
analysis-f0-min 157.2
analysis-f0-max 445.4
```

Remember to delete all existing `desc.mrq` and `.llsm` files once you modify and save `moreconfig.txt`.

In addition, if all samples under a certain directory feature breathy voice or background noise, you may want to add "`analysis-anti-distortion on`"; in the case of background noise, you may also add "`analysis-noise-reduction on`", but this won't help for breathy voices (because breathing noise and background noise, obviously, are two different types of noises).

The way of correcting `desc.mrq` files has been illustrated in Fixing pops/glitches/noises section. This may take some time and effort depending on the length of recordings. If Moresampler already works well with the voicebank configuration, you may skip this step.

After all these procedures, please remove all `.llsm` files when packing and distributing your voicebank, thank you!

Oto generation mode

Since version 0.7.0 Moresampler has the extra feature of being a standalone to automatically generate a fully-labelled `oto.ini` file from sound samples. This novel feature is still under experiment and for now only Japanese continuous-speech voicebanks with file names written in hiragana/romaji are supported.

This feature works on a group of `.wav` files instead of each individual file. Drag a folder containing all the `.wav` files (nested directories won't be counted) onto `moresampler.exe` and Moresampler will take care of the rest.

Please backup existing oto.ini file before using this feature. Otherwise data will be overwritten.

The author has made a video tutorial on Moresampler's oto generation mode.

« PREVIOUS POST

NEXT POST »

---

10 Responses to The Complete Moresampler Tutorial

Nik Walker says:
May 1, 2018 at 1:57 pm

Hey, I tried using some flags with my CV bank and sometimes the flag misses notes. Do you know why this might be?

REPLY

robert Chen says:
February 26, 2018 at 8:38 am

Can you tell me how to get the source code of this application? Also, can this software synthesize Chinese singing voice(In my opinion it's might be okay in theory). There used to be a similar singing voice synthesis engine RUCE, however it doesn't seems to be a complete(or perfect) one (There is a demo on bilibili.com--av number 1488909) . I've got to know that there was some relations between RUCE & Moresampler. So I think that this one is better for Chinese for it's better than RUCE.

I've got some doubt when I read the License.txt file. Shouldn't I modify the source code or turn some libraries of it into other softwares? And I wonder under which license does Moresampler release.

REPLY

robert Chen says:
February 26, 2018 at 8:42 am

P.S I want to run this software on Ubuntu. Is there a binary version of linux? Or how can I get the binary version(e.g. Build from source code).

REPLY

Ib says:
January 10, 2018 at 6:33 pm

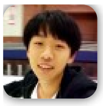
Mr. Hua could there be a way for the u flag being susceptible to tuning, flags, etc. The reason is due to an utau named acme iku, with the u sound she sounds just like her program but i want to be able to tune her while enjoying the u flag.

REPLY

owarase says:
October 8, 2016 at 5:53 pm

Hey, I don't know when it was added(changed?) but I think the 'Resonance' flag should be changed back to just Mf because in practise it doesn't really create/effect any resonance as it simply seems to just 'boost' the 3khz range where as singing resonance is much more complex than just boosting that range.

REPLY



Kanru Hua says:
October 8, 2016 at 6:02 pm

You're right. But since this flag has already been there for 4 months, I'm just leaving it as Mr.

REPLY

TDR says:
August 22, 2016 at 2:54 am

Hello!

I'm just generally curious about the nature of the gender flags. I've been playing around with some formant shifting vsts, and have noticed that even when voice formants shifted downwards in these plugins, spectral graphs are generally shifted downwards, whereas in UTAU through moresampler, there's no spectral shift down. I'm just wondering how you managed that, as I find that rather cool and it definitely gives a much better sound!

Best,
TDR

REPLY

Anon says:
August 17, 2016 at 11:07 am

Got any flag recommendations for using a voicebank with background noise?

REPLY

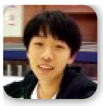
Anon says:
April 21, 2016 at 4:34 am

What to do when you render a ust and there comes some weird noise from certain Parts of an ust with an normal rendered frq map? should I open the frq map in the frq editor in this case?

VB; 春歌ナナ おしゃまさん連続音
frq; _てててぃてた

some more seemed to glitch too but i couldn't find all for now

REPLY



khua5 says:
April 21, 2016 at 8:22 am

https://web.engr.illinois.edu/~khua5/index.php/2016/04/07/the-complete-moresampler-tutorial/#fix-pops

REPLY

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Website

Post Comment

« PREVIOUS POST

NEXT POST »