
ISiS_doc Documentation

Release 1.2

Axel Roebel

Mar 29, 2020

Contents:

1	ISiS: Ircam Singing Synthesis	1
2	Installation	3
2.1	Installing the Software	3
2.2	ISiS Databases	6
3	The ISiS command line	9
3.1	First steps	9
3.2	All command line arguments	11
4	Score files	15
4.1	ISiS scores	15
5	Advanced configuration files	21
6	Manual adaptation of generated parameter contours	23
7	Credits	25
8	Indices and tables	27

ISiS: Ircam Singing Synthesis

The ISiS software is a command line application for singing synthesis that can be used to generate singing signals by means of synthesizing them from melody and lyrics.

The ISiS software is the result of the french national project [ChaNTeR](#) (Project [ANR-13-CORD-011](#)), that was performed in collaboration with [Acapela](#), [LIMSI](#), and [Dualo](#). and is distributed free of charge for members of the IRCAM Forum.

ISiS operates offline synthesis by means of reading score and lyrics from data files and renders the result into an output sound file. In its current version ISiS supports synthesis with 3 French singing voices:

1. **RT**: a tenor male pop singer, and
2. **MS**: a female mezzo-soprano pop singer, and
3. **EL**: a female soprano lyrical singer.

For working with ISiS the software and singing voices need to be installed. Please see detailed instructions for the installation of the command line application and singing voices in the following section

2.1 Installing the Software

ISiS is distributed via the [ISiS page](#) of the [IRCAM Forum](#) as a dmg or tar archive containing a self contained command line application for *MacOS* (\geq El Capitan) or *Linux* (running on all platforms that are supported by [Anaconda Python](#)). While ISiS has been developed in *Python* it comes as a binary executable so that you don't need to install any dependencies besides the ISiS Software and the singing voice packages. The software is currently in a beta stage, it has been tested only on very few systems.

The installation is performed in the following **3 steps** that are rather similar on *MacOS* and *Linux*. Please do not miss to test your installation as described under [testing](#) after you finished with step 3.

- **Step 1) Unpacking:** The unpacking step is slightly different on *Linux* and *MacOS*. For **MacOS**: please locate and double-click the dmg `ISiS_Vx.y.z.dmg` which will mount a disk image containing the application. For **Linux** there are many different file managers that behave slightly different and therefore we will not be able to describe how to extract the distribution from the tar.bz2 file. In most cases you should be able to locate the archive `ISiS_Vx.y.z.tar.bz2` in your file manager (For example Dolphin on KDE or Nautilus on gnome) and either double-click the file or right click the archive and select “extract here” to unpack it.
- **Step 2) Relocating the software:** Move and rename the `ISiS_Vx.y.z` directory you find in the mounted dmg (*MacOS*) or that you extracted from the tar (*Linux*) to a place where you like it to reside.
- **Step 3) Configuring PATHs:** Finally you need to configure executable search path of your terminal so that the `isis` command will be found in the terminal. This can be performed automatically (using a shell script that comes with the ISiS distribution) or manually. The two procedures are described below.

Automatic PATH configuration: The automatic shell script configuration currently supports the use of *bash* (the current default shell under *MacOS*) and *tcsh* shells. In case you use other shells please see the manual configuration instructions and adapt those to the shell you use. Due to very restrictive security policies of Mac OS X the following instructions differ slightly for *MacOS* and *Linux*.

Under MacOS please first **double-click the ISiS_Vx.y.z** folder to open it in the Finder and then open a terminal window. Then back in the Finder window containing the ISiS_Vx.y.z folder grab the **Install_ISiS_commandline.sh** script and drag it with the mouse onto the terminal window. This will paste the complete path to the `Install_ISiS_commandline.sh` into the terminal command line. Once you have copied the path to the terminal please click on the terminal window and then simply hit return.

Under Linux please first **double-click the ISiS_Vx.y.z** folder to open it in your file and directory browser. You can then simply double-click the **Install_ISiS_commandline.sh** script which will execute the script.

The script will configure the terminal such that each time you open a new terminal the `isis` folder is added to your environment. In the following you find the output that is generated for user `vox` installing the ISiS software in his Applications directory on computer `medcomp` running MacOS, the output will be slightly different depending on your setup and OS, but the important line is the second last line telling you that you are set to run the ISiS application from your new installation.

```
medcomp: (~) 501> /Users/vox/Applications/ISiS_V1.2.3/Install_ISiS_commandline.sh
updated /Users/vox/Library/Application Support/Ircam/ISiS_init_rc.sh
updated /Users/vox/Library/Application Support/Ircam/ISiS_init_rc.csh
/Users/vox/.tcshrc is up to date.
/Users/vox/.bashrc is up to date.
/Users/vox/.bash_profile is up to date.
=====
Shell configuration updated to use ISiS from /Users/vox/Applications/ISiS_V1.2.3
=====
```

Manual PATH configuration: Depending on the shell you use you need to open the startup config file of the shell and add the directory ISiS software directory to your *PATH*. To find out which shell you use please open a terminal and type

```
echo $0
```

and hit `<return>`. The shell will display the name of the shell `env` with a leading dash.

Hitting the `<return>` key anywhere in a line in the terminal indicates to the shell that you are finished typing and that the line should now be executed. To avoid extreme redundancy from now on, if you are asked to **execute** a command, then this will mean you should type the command and hit `<return>`.

The shell configuration files are located in your *HOME* directory. The *HOME* directory is the directory where you are located when you are opening a new terminal. You can always go back to the *HOME* directory within your terminal by means of executing

```
cd
```

In the following directory names the *HOME* directory is indicated by means of the `~` sign. This sign is understood by all unix shells, so that you could also go back to your *HOME* directory by means of executing

```
cd ~
```

For editing the files you can for example use the *nano* editor that is available on Linux and MacOS. Please note that while you may use TextEdit on MacOS the file open dialogs will in general not show the shell configuration files.

Assuming again your username is `vox`, and you use *bash* as your shell, and the ISiS version is 1.2.3. In this case the software folder will be named `ISiS_V1.2.3` and if you want to install `ISiS_V1.2.3` into your local Applications directory the path to the software (on MacOS) will be `/Users/vox/Applications/ISiS_V1.2.3`. Accordingly you will need to add the following line to the file `~/.bash_profile`. For this you would execute


```
nano ~/.bash_profile
```

scroll to the bottom, and on the beginning of a new line type

```
export PATH=/Users/vox/Applications/ISiS_V1.2.3:$PATH
```

The export command needs to be located on an individual line. To achieve this you hit <return> at the end of the line. Then you save the file by means of typing ‘control’+’x’ and selecting ‘Y’ when asked to confirm saving the changes. If you have problems operating *nano* you may read the description on changing *PATH* variable under *MacOS* [here](#).

Please note that you may not have a file *~/.bash_profile*, in which case the *nano* command mentioned above will automatically create it.

In the case that you use *tcsh* or *csh* as your shell you would need to edit the file *~/.login* by means of executing

```
nano ~/.login
```

and add the line at the end

```
set path = ( /Users/vox/Applications/ISiS_V1.2.3 $path )
```

again - if you use *tcsh* or *csh* and this file does not exist *nano* will automatically create it.

For other shells please refer to the documentation of the respective shell to see the names of the config files and the commands to be used to extend the *PATH*.

2.1.1 Important Notes

The *PATH* configuration is permanent and you don’t need to repeat it, besides if you install a new version of ISiS or if you move the ISiS_Vx.y.z folder to a different location.

The terminal configuration will be read whenever you open a **new terminal**, so after step 3 you need to open a new terminal to work with the ISiS command.

Under *MacOS* the *PATH* configuration described above will be active only for programs you run from the terminal. Programs started via the Finder - as for example Max/MSP - do not read these configuration files. In case you would like to run ISiS from such a program, you need to add the *PATH* configurations to the runtime environment of the respective software. Please see the documentation of the software to understand how to do this.

In case you would later like to relocate the software package, you can repeat steps 2 and 3 whenever you want. In that case, however, only terminals that are opened after the *PATH* have been configured in Step 3 will contain the new location of the software.

2.1.2 Testing your configuration

Before working with the ISiS synthesis please test whether you have successfully configured your environment. For this please open a new terminal window and execute

```
echo $PATH
```

You should see the `ISiS_Vx.y.z` folder appearing somewhere at the start of the string that is displayed in the terminal. In case you don't see the folder, then please check whether you have correctly carried out the configuration steps. Notably, after manual configuration please check that the corresponding shell configuration files do exist and contain the desired lines.

2.2 ISiS Databases

ISiS singing voices are distributed via the [ISiS page](#) of the [IRCAM Forum](#). Each voice comes as an independent tar archive containing all sound parameters of a single voice. Additionally, each voice archive contains a small script that can be used to configure a voice as ISiS default voice.

Currently there are three voices available (downloads are for Forum members only):

1. **RT**: a tenor male pop singer: ([download](#)), and
2. **MS**: a female mezzo-soprano pop singer ([download](#)), and
3. **EL**: a female soprano lyrical singer ([download](#)).

2.2.1 Install a new ISiS singing voice

For the ISiS program to be able to find the singing databases all the singing voices have to be gathered in a common root directory, which we will call `ISIS_CORPORA`. In the following we will assume you want to use the directory `/Data/ISiS_DB` to collect your ISiS voices, but you are free to choose any other place that better fits your needs. To prepare the installation of the voices you should create the `ISIS_CORPORA` directory. You can do this either from the terminal using the command `mkdir`, or from your GUI (Finder on MacOS and for example KDE/Dolphin or Gnome/Nautilus on Linux). While this directory can be located anywhere, the voices require in the order of 1GB per singing voice, so please be sure that the disk that you select when creating the directory has at least 3GB of space available.

To proceed with the installation of the singing voices please download the different voices and unpack them into the `ISIS_CORPORA` directory. The voice database archives are compatible with Linux and Mac OS. You simply save the archive into the `ISIS_CORPORA` directory and unpack the `tar.bz2` file you just downloaded. On *MacOS* you can simply double-click the file in the Finder. Under Linux the operation to be performed depends on the file manager you use, in some environments you can double-clicking the archive to open it in a dedicated application that you may then use to extract the content to the target directory, other filemanagers may allow right-clicking and selecting something like *unpack here*.

Finally, on MacOS and Linux you can also use the terminal to unpack the voices. For brevity we will not cover this process in detail here.

The name of the folder you create will depend on the software you use for unpacking. For ease of use it is strongly recommended to rename the unpacked voice folder using a short ideally two character long name. In the following we will assume you use the names *RT*, *EL* and *MS*.

Configuring ISiS voices environment

The ISiS software uses two environment variables to find its voices. These two variables can be defined manually or using an automatic script that needs to be executed in the targeted shell environment.

We will first describe the automatic configuration and in the subsequent paragraph the automatic configuration.

Automatic voice configuration

The automatic voice configuration currently supports only *bash* (the current default shell under *MacOS*) and *tcsh* shells. In case you use other shells please see the manual configuration instructions below and adapt those to the shell you use. Due to very restrictive security policies of Mac OS X the following instructions differ slightly for *MacOS* and *Linux*.

Under Mac OS please first double-click the folder containing the data of the specific voice you would like to use as default voice to open it in the Finder and then open a terminal window. Then back in the window containing the *ISiS_Vx.y.z* folder locate the file **Use_as_ISiS_default_voice.sh** and drag it onto the terminal window. This will paste the complete path to the script into the terminal command line. Once you have copied the path to the terminal please click on the terminal window and then simply hit *<return>*.

Under Linux please first double-click the folder containing the voice data to open it in your file and directory browser. You can then simply double-click the **Use_as_ISiS_default_voice.sh** script which will execute the script.

The script will place an ISIS voice config file under your HOME directory. For MacOSX the exact location of the file is

```
~/Library/Application\ Support/Ircam/ISiS_init_voice_rc.cfg
```

for Linux it is located under

```
~/local/AppSupport/Ircam/ISiS_init_voice_rc.cfg
```

In the following you find the output that is generated for user vox installing the RT voice in the */Data/ISiS_DB* directory as the default voice. The two lines are confirming that you are using the */Data/ISiS_DB* directory as voice root and the */Data/ISiS_DB/RT* voice as your default voice.

```
medcomp: (~) 5001> /Data/ISiS_DB/Use_as_ISiS_default_voice.sh
=====
ISiS configuration updated to use </Data/ISiS_DB> as root dir of ISiS voices
and </Data/ISiS_DB/RT> as default voice.
=====
```

Manual voice configuration

You can manually create or change the file that contains the ISIS voice specifications.

For MacOSX the exact location of the file is

```
~/Library/Application\ Support/Ircam/ISiS_init_voice_rc.cfg
```

for Linux it is located under

```
~/local/AppSupport/Ircam/ISiS_init_voice_rc.cfg
```

You can edit this file with arbitrary text editors but need to save it in text mode. Assuming that you placed your CORPORA under the */Data/ISiS_DB* directory and want to use the voice *RT* as your default voice then you would need to create the file with the following content

```
[ISIS_VOICE_CONFIG]
# root directory of ISiS voices
ISIS_CORPORA: /Data/ISiS_DB
# default voice sub directory
ISIS_VOICE: /Data/ISiS_DB/RT
```

Setting voice defaults via environment variables

You can temporarily overwrite the singing voice defaults using two environment variables: the variable *ISIS_CORPORA* defines the root directory containing all ISiS voices and the variable *ISIS_VOICE* defines either the full path or the sub directory under *ISIS_CORPORA* that contains the default voice.

If we assume as before that your voice root directory is */Data/ISiS_DB* and the voice directory containing the voice you want to use as default is */Data/ISiS_DB/RT*, and in case your shell is *bash* then you can execute the following lines in our terminal to switch the voice configuration without changing the config file.

```
export ISIS_CORPORA='/Data/ISiS_DB'
export ISIS_VOICE='RT'
```

You can set only the *ISIS_VOICE* variable if you want to change the default voice without changing the voice root only. These changes will be in effect for all subsequent runs of *isis* you perform in the terminal where you issued these settings. To reset the change use

```
unset ISIS_CORPORA
unset ISIS_VOICE
```

If you don't use *bash* please look into the documentation of your shell to find how you can set environment variables.

The ISiS command line

Once the initial steps are performed you are ready for the first synthesis. For this please open your terminal app and type (where the > character represents the shell prompt and should not be typed)

3.1 First steps

```
> isis.sh -v
```

If all went well, you should receive the isis version displayed on the terminal.

```
ISiS version::1.2.7
```

In case you receive an error message that means that at least one of the steps described so far have not been performed correctly. please check all of them, and if you don't find the problem please contact IRCAM Forum support, sending the following pieces of information

1. The error message you received
2. The content of the .bashrc, .bash_profile, .tcshrc, and .cshrc files you modified
3. The output you receive when you type

```
> echo $SHELL
```

in your terminal.

3.1.1 Synthesising the default song

the default song is a short extract from the French song *Les feuilles mortes*. We will first experiment a little bit with the singing synthesis on the the command line before we will describe in more detail the parameters you can manipulate in the score cfg file.

As a first trial please run the command

```
> isis.sh -o defsong.wav
```

After a about 20 seconds (the time depends on the power of your computer) and a long list of cryptic output the command prompt will reappear. The last lines should display

```
#####
PaN voiced synthesis
#####
#####
PaN unvoiced synthesis
#####
#####
apply post-processing treatments
#####
Create: defsong.wav using wav format!
=====
computed in 28.277118921279907s
```

Which means all went well and you can now listen to the song by means of running

```
> open defsong.wav
```

in the terminal. This will open the default application for wav snd files allowing you to listen to the result. In case you installed MS as default voice you should get [defsong_MS.wav](#)

If you use another singing voice as default you may get a little bit strange sounding results. The score is in fact written for soprano voices and especially if you use RT as singing voice the result suffers from the required transpositions. So depending on the default voice you have configured you should adapt the command line to transpose the default melody, such that it better matches the singing voice.

For MS the database is recorded in 315Hz (approx midi note 63) which is slightly below the average note frequency of the default song. For RT the database is recorded at 150Hz (approx. midi note 50) and to adapt the average note pitch to the voice you can use the command line flag `--global_transp` which transposes the melody by the given number of half tone steps. For RT you get good results by means of lowering the melody by 13 half tones as follows

```
> isis.sh --global_transp -13 -o defsong_RT.wav
```

which will produce the singing which results in [defsong_RT.wav](#)

Finally for EL the database is recorded at about 440Hz (midi note number 69) and to get a good result you could transpose the melody upwards by 2 half tones

```
> isis.sh --global_transp 2 -o defsong_EL.wav
```

which results in [defsong_EL.wav](#)

In case you have established the `ISIS_CORPORA` environment variable you can select the singing voice simply by means of selecting the sub directory in the `ISIS_CORPOPA` folder. To select the MS voice you would simply run the synthesis as follows:

```
> isis.sh -sv MS -o defsong_MS.wav
```

3.1.2 Automagically open synthesized sounds

Before discussing all the different options you can select on the ISiS command line a final tip that simplifies inspecting the synthesis results.

If you add the `-a` flag to the command line

```
> isis.sh -sv MS -o defsong_MS.wav -a
```

the synthesised snd will be loaded into the AudioSculpt application, together with the target pitch contours and the phoneme locations in the synthesized snd creating the following AudioSculpt window

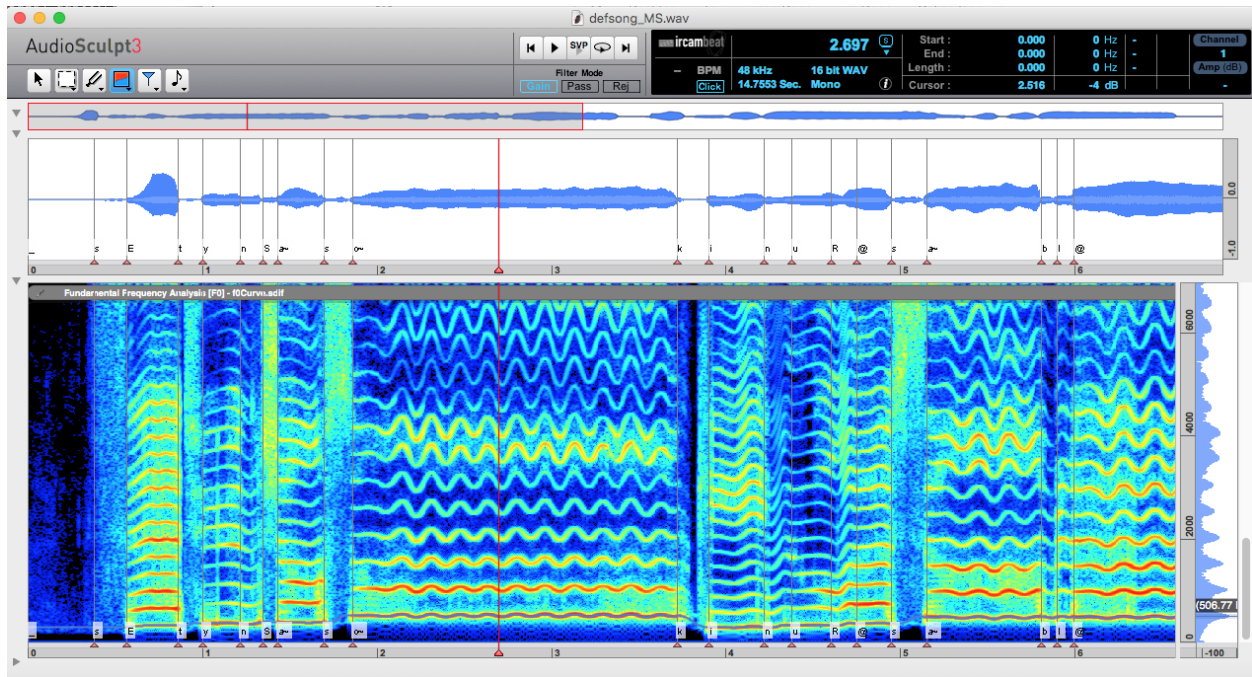


Fig. 1: MS default song displayed in AudioSculpt

If you don't have the AudioSculpt application installed you can use the `-O` flag with a similar purpose, it opens the synthesized sound file in the default application you use, for playing sound files.

```
> isis.sh -sv EL -o defsong_EL.wav -O
```

3.1.3 A complex example

To demonstrate the sound quality that can be obtained with ISiS we use here a fully synthetic extract from the *Opera I.D.* produced by Arnaud Petit

Synthetic mockup from the opera

3.2 All command line arguments

For a discussion of all command line arguments please read

3.2.1 The ISiS command line arguments

Flags and flag variants

Most of the ISiS command line flags have two variants: a short and a long form. The long form is given by means of two dashes “-” while the short form is introduced by a single dash ‘-’. We take as example the flag that requests automatically opening the synthesized sound and the related pitch contour and phoneme sequence. In the introductory section we have learned that this can be achieved by adding `-a` to the command. This is the short form “-a”, and equivalently we could have given the long form `-auto_open` as argument. So the following two commands

```
$> isis.sh -sv EL -o defsong_EL.wav -a
```

```
$> isis.sh -sv EL -o defsong_EL.wav --auto_open
```

are strictly equivalent. In the following we will generally introduce only the long form of the parameter, as for example (`-auto_open`). The long form is supposed to be easier to remember as it declares its function by means of its name. However, in case we want to directly describe both forms we will use a vertical bar to separate both forms (`-a|auto_open`).

Finally, ISiS supports short cutting the long flag to arbitrary shorter lengths, as long as the flag remains unambiguous. So another equivalent command line would be:

```
$> isis.sh -sv EL -o defsong_EL.wav --auto
```

Help

One of the most important flags for ISiS is the help flag (`-h|help`), which can be used to list all command line flags with a short help text.

To see the full list of command line argument of ISiS you can therefore simply run

```
$> isis.sh -h
```

which will display a short usage summary, then the program version, and finally a long help section listing all flags in short and long form, together with default values, that here are given for the case that ISiS version 1.2.5 installed under `/Users/me/Applications/ISiS_V1.2.5`:

The command line arguments are grouped into sections that affect different aspects of the synthesis. We will discuss these sections in the the following paragraphs

ISiS IO

The IO parameters specify input and output files,

- **flag `-melody`**: provides the name of the *ISiS scores* that contains the textual description of melody and phonemes.
- **flag `-outfile`**: the name of the output file. The file format of the output file is AIFF format, however the format will be adapted automatically to the output file extension.

The following sndfile formats will be used depending on the extension.

extension	Fileformat
.au	SUN AU format
.wav	MS WAV
.caf	Apple CAF
.aif/.aiff	SGI AIFF

ISiS Voice

The voice parameters select the singing voice that is used. The singing voices that are currently available for ISiS are described in [the installation instructions](#).

- **flag `--corpora_root`:** overrides the `$ISIS_CORPPORA` environment variable you should have created when installing the singing voices as explained in the section on [singing databases](#). This parameter is not very important if you followed the advice to put all singing voices into the same directory and you have correctly initialized the `ISIS_CORPORA` environment variable (see [configure `ISIS_CORPORA` variable](#)).
- **flag `--singing_voice`:** this flag on the other hand allows you switching voices for each individual synthesis. this flag obtains its default from the `ISIS_VOICE` environment variable that you can position by means of running the install as default voice app in each voice directory.

ISiS Style

Singing style in ISiS can be controlled on various levels. The basic command line style parameters allow selecting basic style models, and as well as controlling certain aspects of the use of style models. A singing style in ISiS is a model that controls the singing parameter contours pitch and intensity, as well as the duration of the different phonemes, depending on the musical context (note frequencies, and note durations). There are 5 preconfigured singing style models available in ISiS: these are eP, jG, fL, sD, and finally None.

The style models have been defined with the help of a musicologist, the None style simply uses default settings of the pitch style parameters, and intensity contours from the singing voice databases.

- **flag `--singing_style`:** selects a singing style model from the five available models that are named: eP, jG, fL, sD, None. The default style is derived from the environment variable `ISIS_STYLE`, and if that is not set, then default style None is used.
- **flag `--no_loudness_style_model`:** exclude note attack and release intensity contours from the effect of the selected style model (Def: False).
- **flag `--no_f0_style_model`:** exclude pitch contour parameters (note attack, release, note transitions, vibrato parameters) from the effect of the selected style model (Def: False)
- **flag `--global_transp`:** note quite the same level as the other styme parameters, here a global transposition in midi notes can be applied to the score (Def: 0)

ISiS opts

A few options processing options are available.

- **flag `--auto_open`:** triggers opening the resulting synthesis file in the AudioSculpt application. The sound file will be opened with phoneme annotations and the generated pitch contour. THE default behavior is to respect the corresponding setting in the `default.configSynth` parameter.
- **flag `--no_auto_open`:** prevents opening the synthesis results result in AudioSculpt even even if the `auto_open` parameter in `default.configSynth.cfg` file is set.
- **flag `--Open`:** open the synthesis result in the default application for sound files (Def: False)
- **flag `--pan_parts`:** output separate voiced and unvoiced singing signal next to the output file (Def: False)
- **flag `--quiet`:** suppress logging of progress messages to the terminal console (Def: False)

ISiS advanced parameters

The following advanced parameter can be used to control more aspects of the singing synthesiser.

TODO: add description*

- **flag -cfg_style:** Singing style configuration file (Def: default.configStyle.cfg)
- **flag -cfg_synth:** Synthesis configuration file
- **flag -styles_root:** root directory for style model files (Def: default.configSytnh.cfg)
- **flag -temproot:** directory for storing temporary intermediate files files (Def: cfg_synth::TEMPFILESPATH)

After having understood the basic options that are available on the command line, we will now discuss the central control of a singing synthesis system: the score. The score gathers all basic melodic and lyric parameters of a singing performance. this comprises the sequence of notes to be played, the tempo, the sequence of phonemes to be sung, as well as note dynamics.

For an in depth discussion of the representation of singing scores in ISiS please read

4.1 ISiS scores

ISiS score files are simple text files with configfile syntax as described in the documentation of python's [config parser module](#).

4.1.1 Configuration file syntax

A configuration file consists of *sections*, each led by a section header marked by means of square brackets, followed by key/value entries separated by a specific string (either = or : may be used). By default, section names are case sensitive but keys are not. Leading and trailing whitespaces are removed from keys and values. Values can also span multiple lines, as long as they are indented deeper than the first line of the value. Empty lines that contain white spaces will be treated as continuation lines of the same sequence of values. Configuration files may include comments, prefixed by specific characters (# or ;). Comments may appear on their own on an otherwise empty line, possibly indented.

With respect to section and key names it is important to understand that when reading a config file, the parser will only consider known headers and keys, all unknown headers and keys will be silently ignored.

4.1.2 Score example

There are two obligatory sections that have to be present in a score: the *[lyrics]* section specifying the lyrics to be sung in terms [XSAMPA](#) phonemes, and the *[score]* section, that contains the melody description.

We will describe the keys that make up the two sections by means of looking into the default score file that is delivered with ISiS and will be used whenever no score has been specified on the command line.

The default score

4.1.3 Lyrics section

The lyrics section contains a single required key termed **xsampa** that contains as value the song lyrics in form of a sequence of phonemes expressed in the phonetic alphabet **XSAMPA**. The specification of the lyrics in form of normal text is not supported due to the many possible pronunciation variants that are possible for a given text.

The ISiS system requires the presence of a phoneme to be sung in the singing database. The list of phonemes that are available in the existing singing databases are sufficient to synthesize all French words, they are the following

Phoneme class	phonemes
vowels	a, e, E, 2, 9, @, i, o, O, u, y, o~, a~, e~, 9~
semi vowels	w, j, H
voiced fricatives	v, z, Z
unvoiced fricatives	f, s, S
voiced plosives	b, d, g
unvoiced plosives	p, t, k
nasals	m, n, N
other	R, l

Find more explanations about the translation of text into XSAMPA on the [xsampa example page](#)

XSAMPA Examples

The following list provides examples for each XSAMPA symbol that is supported in *ISiS*:

Vowels

XSAMPA Symbol	API	french word	list of words	explanation
a	a	la	examples	further reading
e	e	beauté	examples	further reading
E		cèpe	examples	further reading
2	ø	peu	examples	further reading
9	œ	jeune	examples	further reading
@		menu	examples	further reading
i	i	vie	examples	further reading
o	o	réseau	examples	further reading
O		sort	examples	further reading
u	u	fou	examples	further reading
y	y	chute	examples	further reading
o~	~	on	examples	further reading- nasalised
a~	ã	banc	examples	further reading- nasalised
e~	~	bain	examples	further reading- nasalised
9~	œ~	un	examples	further reading - nasalised

Semi-Vowels

XSAMPA Symbol	API	french word	list of words	explanation
w	w	aquarelle	examples	further reading
j	j	bille	examples	further reading
H		huit	examples	further reading

Fricatives

XSAMPA Symbol	API	french word	list of words	explanation
v	v	avion	examples	further reading
z	z	ros	examples	further reading
Z		je	examples	further reading
f	f	feu	examples	further reading
s	s	brossent	examples	further reading
S		chat	examples	further reading

Plosives

XSAMPA Symbol	API	french word	list of words	explanation
b	b	table	examples	further reading
d	d	demain	examples	further reading
g	g	bagues	examples	further reading
p	p	pot	examples	further reading
t	t	compter	examples	further reading
k	k	lac	examples	further reading

Nasals

XSAMPA Symbol	API	french word	list of words	explanation
m	m	maison	examples	further reading
n	n	nez	examples	further reading
N	■	camping	examples	further reading

Others

XSAMPA Symbol	API	french word	list of words	explanation
R		tarte	examples	further reading
l	l	lait	examples	further reading

Online resources

To help you to convert a given french text you can make use of online resources. because online resources directly using XSAMPA are hard to find generally you will have to pass via the [international phonetic alphabet \(IPA\)](#).

Text to IPA

To get the IPA transcription of a given french word you can either use online dictionaries as for example www.wordreference.com, or even transcribe a given phrase completely into IPA using [French to IPA](#).

IPA to XSAMPA

In a second step you can then use the [IPA to SAMPA converter](#) to get the transcription in the SAMPA alphabet, which for the part of the IPA that is covered by *ISiS* corresponds more or less with the XSAMPA implemented in *ISiS*. The only difference are the nasalised vowels that in *ISiS* are always lower-case, while in the SAMPA transcription produced by means of [IPA to SAMPA converter](#) there is a difference between e~ and E~. So when constructing the *ISiS* score you should convert all nasalised vowels, that are letters with a “~” attached, from upper-case into the corresponding lower-case letter.

4.1.4 Score section

The score section needs to intends to provide the information about the melody to be sung. There exist two means to specify the melody. The first one consists of explicitly writing the note information into the score section by means of specifying **midiNotes**, **tempo**, and **rhythm**, as well as an optional **loud_accents** key.

In the near future other score specifications methods will be added that will allow reading the score from midi or MusicXML files.

Explicit notation

The explicit notation of the melody uses the following terms:

midiNotes: the **midiNotes** value specifies the melody to be sung in terms of midi notes. The conversion between midi note number and frequency in Hz is given by

$$\text{Note_hz} = 440.0 * 2.0^{*((\text{midiNote} - 69) / 12)}$$

Accordingly, midi note number 69 represents the note A4 with fundamental frequency of 440Hz, and increasing the note frequency by 1 increases the fundamental frequency by one half tone. Note, that in an extension of the normal midi notation *ISiS* note numbers are not constrained to be integers. This means that quarter note intervals can be represented easily by means of using 0.5 steps, that means the quarter tone above A4 would be denoted as midi note 69.5.

tempo : the **tempo** value defines the BPM, that is in form of the number of quarter notes per minute. The length of a single quart note in seconds is then

$$\text{nl_s} = 60 / \text{tempo}$$

rhythm: the **rhythm** value determines the length of all the notes in the **melody** section. The note length is specified in quarter notes, which means that noting a one here will produce a note of the length of a quarter note.

loud_accents: the **loud_accents** value can be used to control the dynamic of individual notes. In cse no loudness accents are specified the note loudness will be obtained from the singer database, where all singers have sung with approximately constant loudness. The same result will be obtained if for each note an loudness accent value of

1 is provided. In case you would like to increase or decrease the loudness of individual notes, you can specify arbitrary positive value as loudness accents, that are interpreted as factors to be applied to perceived loudness, such that specifying a loudness accent of 2 for an individual note will render the note approximately twice as loud.

Reading the score from a midi file

Attention: this functionality is not yet finalized and still requires implementation and testing

Reading the score from a MusicXML file

Attention: this functionality is not yet finalized and requires implementation and testing

CHAPTER 5

Advanced configuration files

ToDo

Manual adaptation of generated parameter contours

ToDo

CHAPTER 7

Credits

Particular thanks for contributions go to

- Luc Ardaillon, for having worked on the ISIS software during his [PhD thesis](#) in developing a large part of the software and the singing style models,
- Marlene Schaff, [Raphaël Treiner](#), and other singers for contributing their voices.
- [Acapela](#) for contribution of the annotation of the singing corpora,
- All participants of the [ChaNTeR](#) project for valuable discussions throughout the project,

CHAPTER 8

Indices and tables

- search