

SpeechRecorder Quick Start and User Manual

Christoph Draxler

draxler@phonetik.uni-muenchen.de

Klaus Jänsch

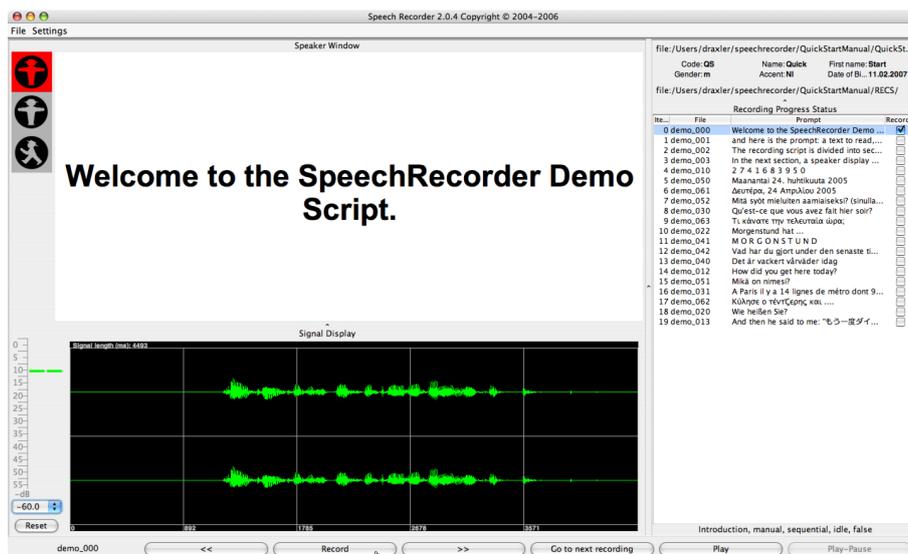
klausj@phonetik.uni-muenchen.de

Institut für Phonetik und Sprachverarbeitung

Universität München

April 13, 2015

April 13, 2015



SpeechRecorder is an application for script-driven speech, audio, and signal recordings. Its main features are

- platform independence
- automatic and manual recording progress
- number of recording channels dependent only on the audio hardware
- speaker and supervisor views on multiple screens
- full Unicode text, image and audio prompts

Quick Start

SpeechRecorder organizes recordings in projects. A *project* is a combination of a speaker database, a set of recording scripts, and a set of recording sessions. A recording session consists of an individual speaker, a recording script, the selected recording settings, and a directory into which the recorded files are written.

1. Download *SpeechRecorder* from <http://www.speechrecorder.org>. Java Web Start on your machine should automatically start *SpeechRecorder*.
2. Select the command **Project > New** from the menu, give the project a name. The following items will now be created:
 - a workspace directory 'speechrecorder' in your home directory
 - a project directory in the workspace directory
 - a sample (or empty) recording script
 - an empty speaker database
 - a project configuration file

On the left side of the display, a small traffic light will show up. In the middle, the prompt area is displayed, and on the right side, the contents of the recording script are listed (see fig. 1 a)).

3. In the **Speakers** menu, select the option **Speaker . . .** and enter data for a speaker. Select the speaker in the table and close the dialog with the button **select**.
4. Click the button **Record** to start your recording session. Stop the current recording by clicking **Stop** or waiting until the recording timeout has been reached. After the recording has ended, the signal is displayed on the screen. Click on **Play** to listen to the recording.
5. Proceed to the next item by clicking **>>**. Start the next recording with the **Record** button.
6. After the final item has been recorded, *SpeechRecorder* displays a message. Click **Ok** to acknowledge the message.
7. You will find your recordings in the subdirectory **RECS** of the project directory. The menu item **Info** in the **Help** menu will help you to find the project file paths.

You're done – you've recorded your first session using *SpeechRecorder*!

Demo Script

The first section contains test items, recording progress is manual (i.e. you have to click to start and end a recording, and click again to proceed to the next item), and only the supervisor view is shown. The second section contains sample prompts in different languages and of different types. Recording progress is semi-automatic (i.e. after a recording is stopped, the script proceeds automatically to the next item), and the speaker view is displayed.

Multiple Displays

If you have two displays attached to your machine, the supervisor view will always be shown on the primary display, the speaker view on the secondary display(s) (see fig. 1 a) and b)).

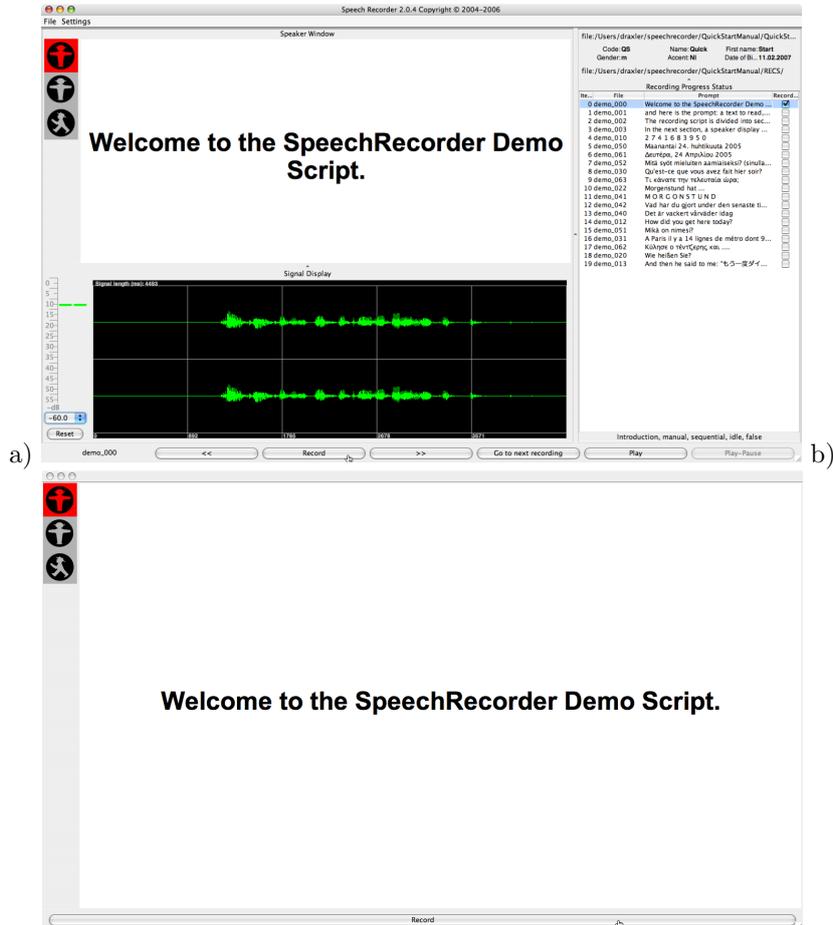


Figure 1: SpeechRecorder supervisor (a) and speaker (b) views

Citing SpeechRecorder

SpeechRecorder was originally presented at the International Conference on Language Resources and Evaluation in Lisbon in 2004 [?]. Please use this reference to *SpeechRecorder* in your publications.

Contents

1	Recording Script	5
1.1	Edit recording script with internal script editor	5
1.2	Edit recording script with internal or external XML editor	6
1.3	The <section> element	6
1.4	The <recording> element	7
1.5	The <recprompt> element	9
1.6	The <mediaitem> element	9
2	Recording Phases	11
3	Menu File	11
4	Menu View	12
5	Menu Project	12
6	Menu Speaker	13
7	Menu Script	13
8	Menu Settings	14
9	Recordings via the Internet	14
10	Miscellaneous	14
A	Recording script DTD	15
B	Reserved keywords for recording scripts	16
C	Known issues	16
C.1	Platform dependencies	17
D	Signal quality problems	18
E	Contacts and Copyright	18
F	Useful links	19

1 Recording Script

A script specifies which items are to be recorded. A *script* consists of two parts, a header containing meta-data items, and the recording script proper. The *recording script* is divided into sections. A *section* is an organizational unit that specifies the presentation order, and progress mode for the recording items it contains.

A *recording item* or *recording* consists of the instructions, the prompt item, and a comment. Instructions and comment are optional. A prompt item consists of text, an image, or an audio clip. The text may be stored in the recording script, or fetched from an external file or URL. Images and audio clips must be loaded from external sources, e.g. a file or a URL.

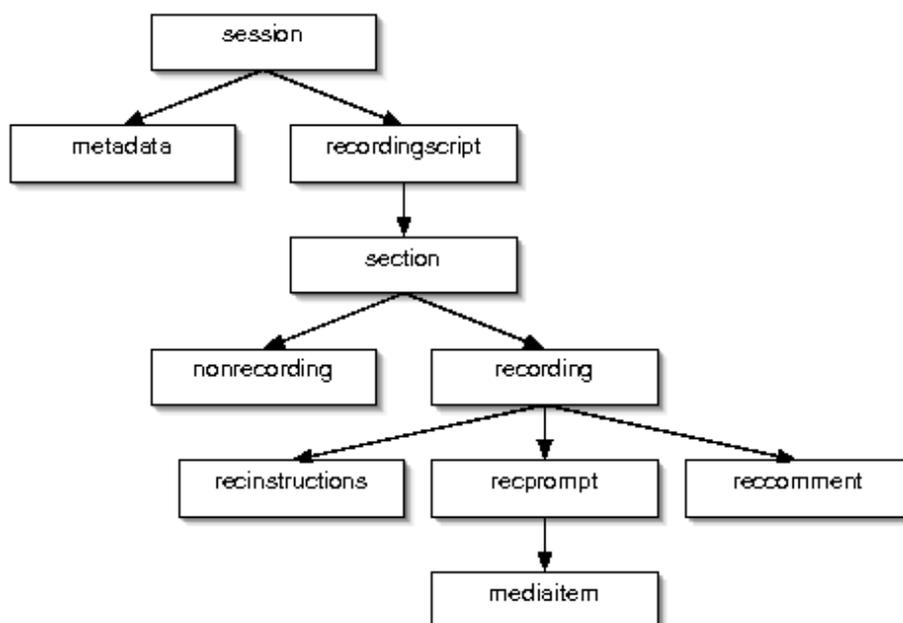


Figure 2: Structure of a *SpeechRecorder* recording script

A recording script is stored as an XML document. The DTD is given in Appendix A.

1.1 Edit recording script with internal script editor

Open the recording script editor with the menu item **Edit script...** in the **Script** menu. The editor has its own menu bar. The **Script** menu contains items to add new sections or recording items. In the left column you can edit recording script name, optional metadata and the list of sections. The middle column represents the selected section and displays the list of recording items. The right column represents the selected recording item. It has two tabs: In the **Prompt** tab you can edit the prompt which will be displayed to the subject. The **Control** tab defines how recording will be controlled during the recording session.

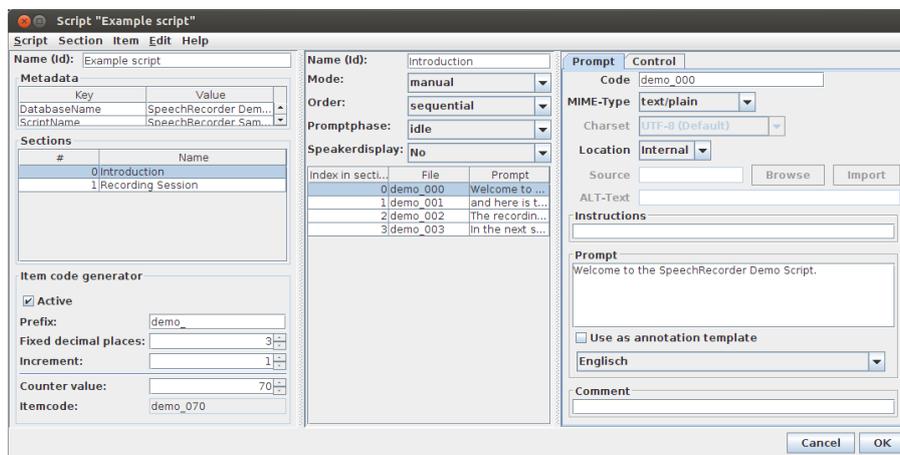


Figure 3: Recording script editor

Selected sections or items can be edited using cut, copy and paste commands shown in the edit menu or by key strokes. If you copy and paste a section or prompt items an active item code generator will apply new item codes to this items to avoid duplicate item codes.

1.2 Edit recording script with internal or external XML editor

If you have some experience using XML files you can edit the recording script with the internal editor (**Edit script source...** menu item in **Script**) or an external XML or text editor.

1.3 The <section> element

A *section* groups together items that are presented and recorded in a similar manner.

In a recording script, the <section> tag is defined as follows:

```
<!ELEMENT section (nonrecording | recording)+ >
```

```
<!ATTLIST section name CDATA #IMPLIED
                speakerdisplay CDATA #IMPLIED
                order CDATA #IMPLIED
                mode CDATA #IMPLIED
                promptphase CDATA #IMPLIED >
```

All attributes are optional. *name* specifies the name of the section, e.g. *Introduction* or *Narrative*. *speakerdisplay* indicates whether the speaker view will be shown or not – allowed attribute values are *yes* and *no*.

order specifies the order in which the items in this section will be presented. The allowed values are *sequential* or *random*.

`mode` controls the recording progress. The attribute value *manual* means that the user has to click once to advance to the next recording item, and again to start the recording. *autoprogess* means that the user clicks only once to advance to and immediately start the next recording. *autorecording* finally means that the script proceeds to the next item and starts its recording without user action. However, the user may pause the script and resume recording later.

`promptphase` specifies when the prompt item is displayed. *idle* displays the item already before the actual recording, e.g. to give the user time for preparation. *recording* shows the prompt only during the recording phase (see section 2 for details, and Appendix ?? for problems when using audio or video prompts).

Sample sections

```
<section name="Introduction" order="sequential"
        speakerdisplay="no" mode="manual"
        promptphase="idle">
...
</section>

<section name="Recording Session" order="random"
        speakerdisplay="yes" mode="autoprogess"
        promptphase="idle">
...
</section>
```

Section display

Information on the section is displayed below the table with the recording items in the supervisor view (fig. 4).

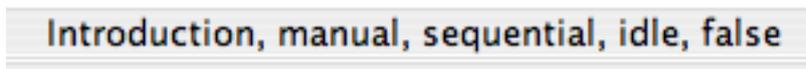


Figure 4: Section information display in the supervisor view

1.4 The <recording> element

The <recording> element defines the id, contents, and timing of the current recording item. It consists of the optional <recinstructions> and <recomment> elements, and the mandatory <recprompt> element. <recinstructions> and <recomment> simply contain text – which is displayed to both the speaker and the supervisor, or the supervisor only, respectively.

```
<!ELEMENT recording (recinstructions?, recprompt, recomment?) >

<!ATTLIST recording itemcode CDATA #REQUIRED
recduration CDATA #IMPLIED
```

```

prerecdelay CDATA #IMPLIED
postrecdelay CDATA #IMPLIED
finalsilence CDATA #IMPLIED
beep CDATA #IMPLIED
rectype CDATA #IMPLIED
blocked CDATA #IMPLIED
annotationTemplate CDATA #IMPLIED
>

```

`<recinstructions>` may have the attributes `mimetype` and `src` to allow instructions to be read in from an external source (see C for details).

The attribute `itemcode` of `<recording>` is mandatory. It uniquely identifies a recording item. `itemcode` can be an arbitrary string – however, because the `itemcode` becomes part of the audio file name, it may not contain characters that have a special meaning in the file system (see C for details). In the script editor GUI you can activate the media item generator, which generates unique item codes for you. The generator consists of a counter and a formatter. If a new recording is added the counter will be incremented by the value configured until the formatted item code value reaches a string value which does not exist in the script. The formatted value is concatenated by the given optional prefix (default "item") and the counter value as a string with fixed decimal places. The settings of the item code generator in the script editor are not stored to the project configuration. If you want to store this settings to the project configuration, open the corresponding form in the project configuration editor (Project -> Preferences ...-> Prompting -> Script).

`recduration` specifies the recording time in milliseconds. The default is to record infinite. `prerecdelay` and `postrecdelay` specify in milliseconds a time span during which recording is active, but the the prompt is still inactive (see 2 for details).

The `finalsilence` attribute is reserved for silence detection. Silence detection is currently not implemented, the attribute has no effect!

The only `rectype` currently supported is *audio*. Default is *audio*.

The `blocked` attribute is of boolean type and determines if playback of a media prompt (i.e. a audio clip) blocks recording. If `blocked` is set, recording starts when playback of the prompt has finished. If not set playback of the prompt and recording start at the same time.

Recording sample

```

<recording prerecdelay="2000"
           recduration="20000"
           postrecdelay="500"
           itemcode="demo_001">
  <recprompt>
    ...
  </recprompt>
</recording>

```

1.5 The <recprompt> element

The <recprompt> element holds the prompt item. It holds one or two media items, which will be prompted to the user.

Example: Simple text prompt.

```
<recprompt>
  <mediaitem>Welcome to the SpeechRecorder Demo Script.</mediaitem>
</recprompt>
```

All text and image media types can be combined with audio prompts. Such a combined prompt will display the text or image and play an audio prompt simultaneously. In the script editor press **Item** *i* **Add extra media item** to add a second media item. In the prompt item viewer a new tab 'Media 02' will appear. Apply the audio file to the new media item.

Examples for combined audio prompts in XML script source:

```
<recprompt>
  <mediaitem>Welcome to the SpeechRecorder Demo Script.</mediaitem>
  <mediaitem mimetype="audio/x-wave" src="resources/audio/welcome.wav"/>
</recprompt>

<recprompt>
  <mediaitem mimetype="image/svg+xml" src="resources/image/welcome.xml">
  <mediaitem mimetype="audio/x-wave" src="resources/audio/wav/welcome.wav"/>
</recprompt>
```

1.6 The <mediaitem> element

The <mediaitem> element describes a media item. It may be empty, or contain text which is displayed on the screen.

```
<!ELEMENT mediaitem (#PCDATA)*>

<!ATTLIST mediaitem mimetype CDATA #IMPLIED
charset CDATA #IMPLIED
src CDATA #IMPLIED
alt CDATA #IMPLIED
autoplay CDATA #IMPLIED
modal CDATA #IMPLIED
width CDATA #IMPLIED
height CDATA #IMPLIED
volume CDATA #IMPLIED
>
```

All attributes are optional.

mimetype specifies the type of prompt item. For image and audio prompts, this attribute provides a hint for displaying the prompt item – image items are drawn on the screen, audio is played via the system speakers or a headphone. Default MIME type is text/plain. **charset** Charset of external text prompt file. The encoding of prompt text embedded in the recording script is inherited from the encoding of the entire recording script which is by default UTF-8. **src** is a file name or a URL from which a prompt item is retrieved.

`alt` contains the text that is displayed if the item cannot be retrieved from the external source.

`autoplay`, `modal` and `volume` apply only to time-dependent prompt items, i.e. audio clips. If `autoplay` is set to *yes* the clip plays automatically as soon as the item is displayed, otherwise the user has to start playback explicitly. With `modal` set to *yes* item playback cannot be interrupted, and `volume` determines the audio volume for playback.

`width` and `height` specify the width and height in pixels of the image or video to display.

It is up to the recording script author to set the `mediaitem` attribute values to meaningful values. *SpeechRecorder* accepts the combinations given in table 1.6.

mimetype	src	alt	autoplay	modal	width	height	volume
text/plain	-	-	-	-	-	-	-
text/html	URL	-	-	-	-	-	-
text/rtf	URL	-	-	-	-	-	-
image/jpeg	URL	+	-	-	+	+	-
image/svg+xml	URL	+	-	-	+	+	-
audio/x-wave	URL	+	+	+	-	-	+

Table 1: Meaningful `<mediaitem>` attribute combinations.

An audio `<mediaitem>` element without contents displays a generic symbol for audio playback. An audio `<mediaitem>` element combined with text or image media displays the text or image contents and plays back the audio.

`<mediaitem>` sample

The following `<mediaitem>` element displays a text prompt:

```
<mediaitem mimetype="text/plain">
  Welcome to the SpeechRecorder Demo Script.
</mediaitem>
```

This `<mediaitem>` element shows a formatted text loaded from a file:

```
<mediaitem mimetype="text/rtf"
  src="promptText.rtf" />
```

Note: RTF prompts only work on Windows operating systems.

This `<mediaitem>` element shows an image loaded from a relative (to the project directory) URL:

```
<mediaitem mimetype="image/jpeg"
  src="images/FelixWas.jpg"
  alt="Boy and washing machine" />
```

This `<mediaitem>` element shows an image loaded from an absolute URL:

```
<mediaitem mimetype="image/jpeg"
  src="http://www.speechrecorder.org/prompts/images/FelixWas.jpg"
  alt="Boy and washing machine" />
```

2 Recording Phases

Each recording is performed as a sequence of phases. The sequence of phases is shown in fig. 5.

A modal prompt display means that the prompt item is shown, but marked as inactive, e.g. by using greyed-out text, low resolution images or a disabled audio button. The default setting is to have modal prompt display during the prerecording and postrecording phases, and an active prompt display during recording. The attribute `promptphase` of a `<section>` element determines the start of an active prompt display, and it overrides the default setting.

IDLE no recording, red light, prompt item is only displayed if the attribute `promptphase` is set to *idle*.

PLAYPROMPT the prompt is playing.

PRERECORDING recording, yellow light, modal prompt item display.

RECORDING recording, green light, active prompt item display.

POSTRECORDING recording, yellow light, modal prompt item display.

A prerecording phase is useful to either record environment noise prior to the main recording, or to give the speaker a precisely delimited time to prepare the utterance. A postrecording phase is most commonly used to avoid signal truncation due to clicking the `stop` button too early.

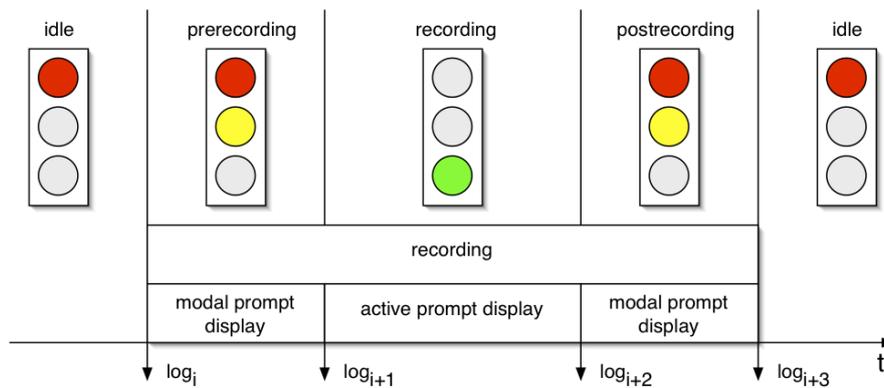


Figure 5: Recording phases

The timing for time-dependent prompts has to be set to appropriate values by the script author, e.g. to make sure that the recording time is sufficient for prompt playback and recording.

3 Menu File

Contains commands to save (configuration) files, to print a screenshot of current session and to quit.

4 Menu View

Contains commands to toggle the speaker display on and off and to customize the audio signal view.

5 Menu Project

Contains commands to create new projects, open a project from the list of known projects or close a project. These projects must reside in the *SpeechRecorder* directory in the user's home directory. A project can only be opened if no other project is open. The **Export** command packs a project archive in a zip-archive file. The **Import** command unpacks a project in a zip-archive file. The project will be deployed in the project directory.

The **Preferences...** command opens a dialog window with the tabs **Project**, **Speakers**, **Recording**, **Playback**, **Control**, **Prompting** and **Logging**.

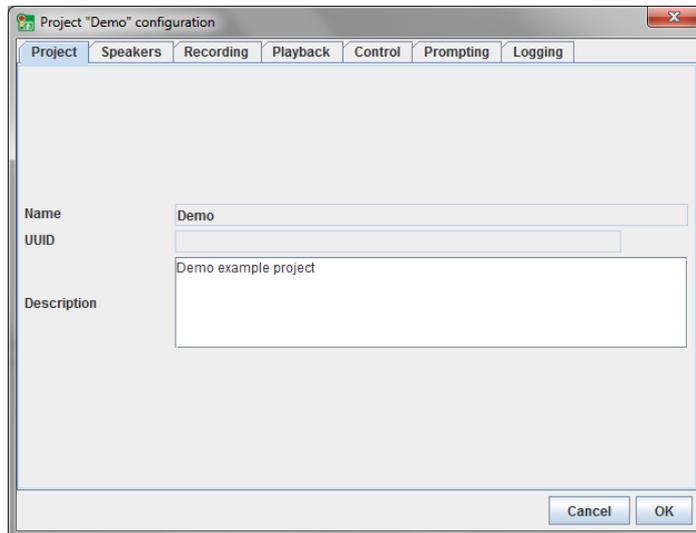


Figure 6: Settings > Project... command

The **Project** tab (fig. 6) presents the project name, optional UUID and optional description. The purpose of the UUID is to distinguish different projects with the same name in the future.

The **Speakers** tab contains the location of the speaker database. This database can be stored in the project directory, or any other accessible location in the local file system.

The **Recording** tab allows the user to set the recording parameters. The tab has the subtabs **Device**, **Channel routing**, **Format** and **Options**. In the **Device** tab (fig. 7) you can select the recording device *SpeechRecorder* should use for this project. The **Channel routing** tab it is possible to specify arbitrary connections between recording device channels and output file channels. This is only necessary for advanced recording setups. The sample rate, quantization, byte order, encoding, number of channels can be chosen in the **Format**

tab. In the **Options** tab you can set parameters whether repeated recordings of an item overwrite previous ones or are stored as versions, the default progress mode, resetting the level meter for every recording, default values for pre- and postrecording phases, the location for the recorded audio files and advanced options for seamless recording in autorecord mode and the recording target. Recording target **DIRECT** stores the captured audio data directly to the final audio (wav) file. **TEMP_RAW** stores first to a temporary file and if recording has stopped converts the audio data to the final audio file. **TEMP_RAW** is recommended if you store your recordings to a network file system if the throughput is not sufficient to store realtime audio data.

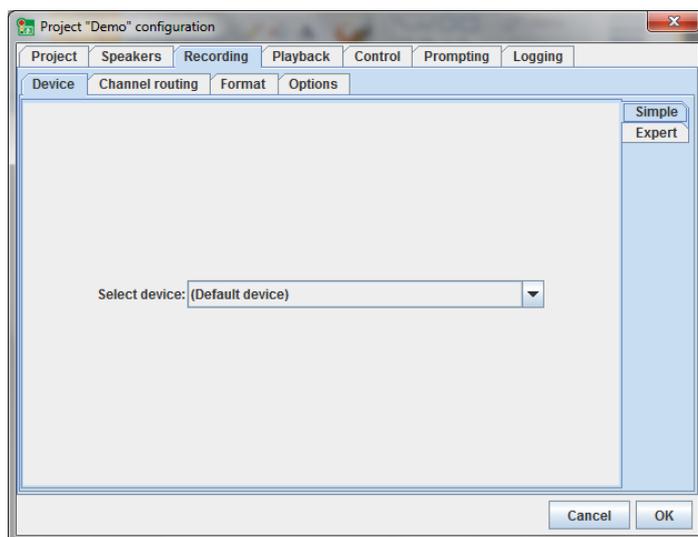


Figure 7: Settings > Project... > Recording > Device dialog window

The **Prompting** tab contains other tabs to edit the type of start stop signal (the traffic light), select fonts, their style and size for prompt, instructions and comment texts, the location of the recording script file, default value for prompt autoplay (audio prompts), customizing the speaker window, project settings for the script item generator and playback device settings for audio prompts.

6 Menu Speaker

The command **Speakers...** opens the speaker database.

7 Menu Script

Contains commands to open the recording script with a editor or to edit the XML source of the script.

8 Menu Settings

The `Skip...` command can be used to skip to a given recording item. The `Recording...` command opens an audio mixer GUI (but we recommend to use the audio mixer of the operating system).

9 Recordings via the Internet

One of the most interesting features of *SpeechRecorder* is its ability to transfer audio files to a remote server. This is achieved using the `http` (*hypertext transfer protocol*) in combination with the `post` method for sending data from the client to a server.

We host some recording projects on our `http://webapp.phonetik.uni-muenchen.de/wikispeech/`:WikiSpeech database system.

```
@InProceedings{ips_bibtex_4966,
author = {Draxler, Chr. and J\"ansch, K.},
title = {{W}iki{S}peech -- {A} {C}ontent {M}anagement {S}ystem for {S}peech {D}atabases},
booktitle = {Proc. of Interspeech},
year = {2008},
address = {Brisbane}
}
```

Please contact us, if you are interested in starting a WikiSpeech project.

10 Miscellaneous

SpeechRecorder logs its activities into plain text log files. The number of log files is dependent on the platform.

A Recording script DTD

```
<!ELEMENT session (metadata?, recordingscript)>
<!ELEMENT script (metadata?, recordingscript)>

<!ATTLIST session id CDATA #REQUIRED>
<!ATTLIST script id CDATA #REQUIRED>

<!ELEMENT metadata (key, value)+>

<!ELEMENT properties (key, value)+>

<!ELEMENT key (#PCDATA)>

<!ELEMENT value (#PCDATA)*>

<!ELEMENT recordingscript (section)*>

<!ATTLIST section name CDATA #IMPLIED
               speakerdisplay CDATA #IMPLIED
               order CDATA #IMPLIED
               mode CDATA #IMPLIED
               promptphase CDATA #IMPLIED
>

<!ELEMENT section (nonrecording | recording)*>

<!ELEMENT nonrecording (presenter?, mediaitem+)>

<!ATTLIST nonrecording duration CDATA #IMPLIED>

<!ELEMENT recording (recinstructions?, recprompt, recomment?) >

<!ATTLIST recording itemcode CDATA #REQUIRED
recduration CDATA #IMPLIED
prerecdelay CDATA #IMPLIED
postrecdelay CDATA #IMPLIED
finalsilence CDATA #IMPLIED
beep CDATA #IMPLIED
rectype CDATA #IMPLIED
blocked CDATA #IMPLIED
annotationTemplate CDATA #IMPLIED
>

<!ELEMENT recinstructions (#PCDATA) >

<!ATTLIST recinstructions mimetype CDATA #IMPLIED
```

```

charset CDATA #IMPLIED
src CDATA #IMPLIED
>

<!ELEMENT recprompt (presenter?, mediaitem+)>
<!ATTLIST recprompt target CDATA #IMPLIED>

<!ELEMENT presenter (properties?)>
<!ATTLIST presenter type CDATA #IMPLIED
classname CDATA #IMPLIED
>

<!ELEMENT recomment (#PCDATA)>

<!ELEMENT mediaitem (#PCDATA)*>

<!ATTLIST mediaitem mimetype CDATA #IMPLIED
charset CDATA #IMPLIED
src CDATA #IMPLIED
alt CDATA #IMPLIED
autoplay CDATA #IMPLIED
modal CDATA #IMPLIED
width CDATA #IMPLIED
height CDATA #IMPLIED
volume CDATA #IMPLIED
>

```

B Reserved keywords for recording scripts

A recording script may contain the following keywords for recording progress, presentation order, and recording type. Recording progress and presentation order are defined via attributes of the tag `<section>`, recording type via an attribute of `<recording>`, and mime-types via `<mediaitem>`.

recording progress: attribute `mode`, values *manual*, *autoprogess*, *autorecord-ing*.

presentation order: attribute `order`, values *sequential*, *random*.

recording type: attribute `rectype`, values *audio*, *video*. Note: *video* as a recording type is not yet implemented.

mime-types: *text/plain* for text, *audio/x-wave*, *audio/x-aiff* for audio, *image/jpeg*, *image/gif* for images.

C Known issues

The following list contains some of the known problems of *SpeechRecorder*. If you find further bugs and errors, please contact draxler@phonetik.uni-muenchen.de.

- When recording with a laptop please disconnect the power adaptor and record using the battery only. External power adaptors may interfere with the audio signal, causing strong distortions. The effect may vary with the brand and type of laptop.
- If you want to use an external XML editor to edit the recording script:
 - Windows: Windows 7 Standard Editor, jEdit (jedit.sourceforge.net) or Atom Editor(atom.io)
 - Mac: we recommend to use the free editor TextWrangler (www.barebones.com) which allows you to import and export content in many encodings.
 - Linux: vim,gedit,jedit,Atom,...
- The following attributes are defined in the recording script DTD, but have not yet been implemented in *SpeechRecorder*:
 - `mimetype` and `src` attributes for the `<recinstructions>` element are ignored.
 - Silence detection to stop a recording is not yet implemented.
 - Recording/prompting video is not yet implemented.
- The directory name into which the audio files are saved is named after the speaker number in the speaker database. Currently, this number is not visible in the speaker database.
- Attribute values for `itemcode` may be arbitrary strings, and the value becomes part of the audio file name. This may cause problems if the string contains characters that have a special meaning for the file system, e.g. `"/`, `":`, `";` etc. It is thus recommended to use only the characters `a-z`, `A-Z`, `_` and `0-9` for the `itemcode` attribute.

C.1 Platform dependencies

- *Mac OS X*: Audio Interface CoreAudio only supports sample rates supported directly by the hardware. Use 'JavaSound' interface for other sample rates on Mac OS X later than 10.7.
- *Windows XP,?*: If the built-in USB audio driver is used, the sample rate will be set to 22.05 kHz when a Windows system beep is output via an M-Audio mobile pre USB device. *SpeechRecorder* does not detect this change of sample rate. All subsequent recordings will be made using the new sample rate. **Solution**: make sure to install and select the most recent M-Audio device driver from <http://de.m-audio.com/>.
- *Windows 7,8*: The operating system distinguishes audio devices of the same model by inserting a counter number in the device name. This only happens if you connect the same device to different (USB) ports. It is recommended to always connect to the same USB port.

D Signal quality problems

If you encounter problems with the signal quality please check your setup.

- When recording with a laptop computer, please disconnect the power cord and run the computer on batteries only. The power supply of many laptops is not well shielded, resulting in strong interferences in the signal (see 8).

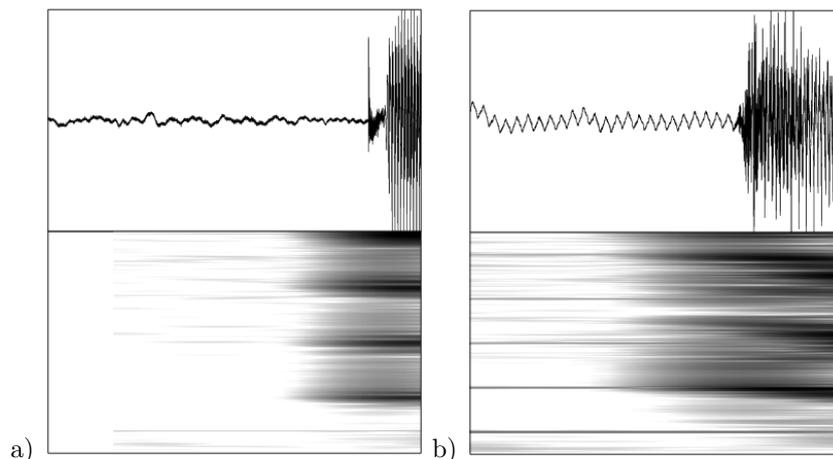


Figure 8: Oscillogram and sonagram of an audiosignal, recorded on a laptop running a) on batteries and b) connected to the power mains. The right oscillogram shows a marked sawtooth-pattern, and the sonagram shows thin horizontal lines at multiples of the mains frequency.

E Contacts and Copyright

SpeechRecorder is being developed by the Institute of Phonetics and Speech Processing of Ludwig-Maximilian University in Munich, Germany. Its main authors are Christoph Draxler and Klaus Jänsch. Many people have contributed to the software by providing localized versions of the graphical user interface, or by suggesting improvements to the software.

The software is Copyright ©2004-2015 by Ludwig-Maximilian University of Munich, Germany. This program comes with ABSOLUTELY NO WARRANTY.

Please cite the original article describing *SpeechRecorder* if you refer to the software in your publications: [?]. Alternatively, you may use the following BibTeX-formatted record:

```
@inproceedings{DraxlerJaensch2004,  
  Author = {Chr. Draxler and K. J{"a"}nsch},  
  Title = {{SpeechRecorder -- a Universal Platform Independent  
          Multi-Channel Audio Recording Software}},  
  Booktitle = {Proc. of LREC},
```

Pages = {559-562},
Address = {Lisbon},
Year = {2004}}

You may use the software free of charge for academic, research and development, and commercial purposes. We particularly encourage the use of *SpeechRecorder* in university or school courses on speech recording.

You may distribute the software freely, provided that the packed `.jnlp` or `.jar` files are not altered.

The software is provided as is. The authors and Ludwig-Maximilians University cannot be held responsible for any damage caused by the use of the software.

F Useful links

- <http://www.phonetik.uni-muenchen.de/Bas/software>: BAS pages with links to software and updates
- <http://java.oracle.com>: Java web pages at Oracle