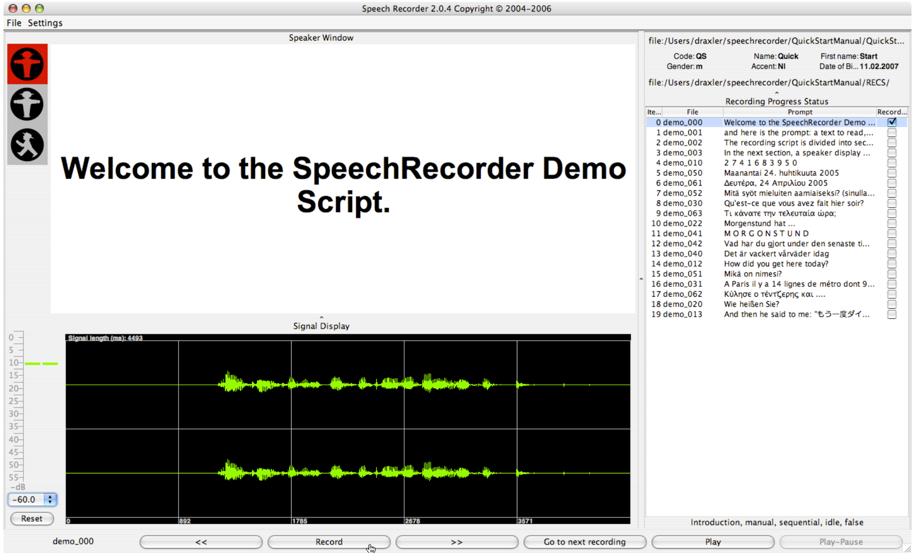
SpeechRecorder Quick Start and User Manual

Christoph Draxler
`draxler@phonetik.uni-muenchen.de`
Institut für Phonetik und Sprachverarbeitung
Universität München



file:/Users/draxler/speechrecorder/QuickStartManual/QuickSt...

Code: **QS** Name: **Quick** First name: **Start**
Gender: **m** Accent: **Nl** Date of Bi... **11.02.2007**

file:/Users/draxler/speechrecorder/QuickStartManual/RECS/

Recording Progress Status

Ite...	File	Prompt	Record...
0	demo_000	Welcome to the SpeechRecorder Demo ...	☑
1	demo_001	and here is the prompt: a text to read,...	☐
2	demo_002	The recording script is divided into sec...	☐
3	demo_003	In the next section, a speaker display ...	☐
4	demo_010	2 7 4 1 6 8 3 9 5 0	☐
5	demo_050	Maanantai 24. huhtikuuta 2005	☐
6	demo_061	Δευτέρα, 24 Απριλίου 2005	☐
7	demo_052	Mitä syöt mieluiten aamiaiseksi? (sinulla...	☐
8	demo_030	Qu'est-ce que vous avez fait hier soir?	☐
9	demo_063	Τι κάνατε την τελευταία ώρα;	☐
10	demo_022	Morgenstund hat ...	☐
11	demo_041	M O R G O N S T U N D	☐
12	demo_042	Vad har du gjort under den senaste ti...	☐
13	demo_040	Det är vackert vårväder idag	☐
14	demo_012	How did you get here today?	☐
15	demo_051	Mikä on nimesi?	☐
16	demo_031	A Paris il y a 14 lignes de métro dont 9...	☐
17	demo_062	Κύλησε ο τέντζερης και	☐
18	demo_020	Wie heißen Sie?	☐
19	demo_013	And then he said to me: "もう一度ダイ...	☐

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

- platform independence
- automatic and manual recording progress
- local and remote recordings via the Internet
- 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 `File > New` from the menu and give the project a name. The following items will now be created:
   - a project directory in your home directory
   - a sample 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 `Settings` menu, select the option `Speakers` 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.

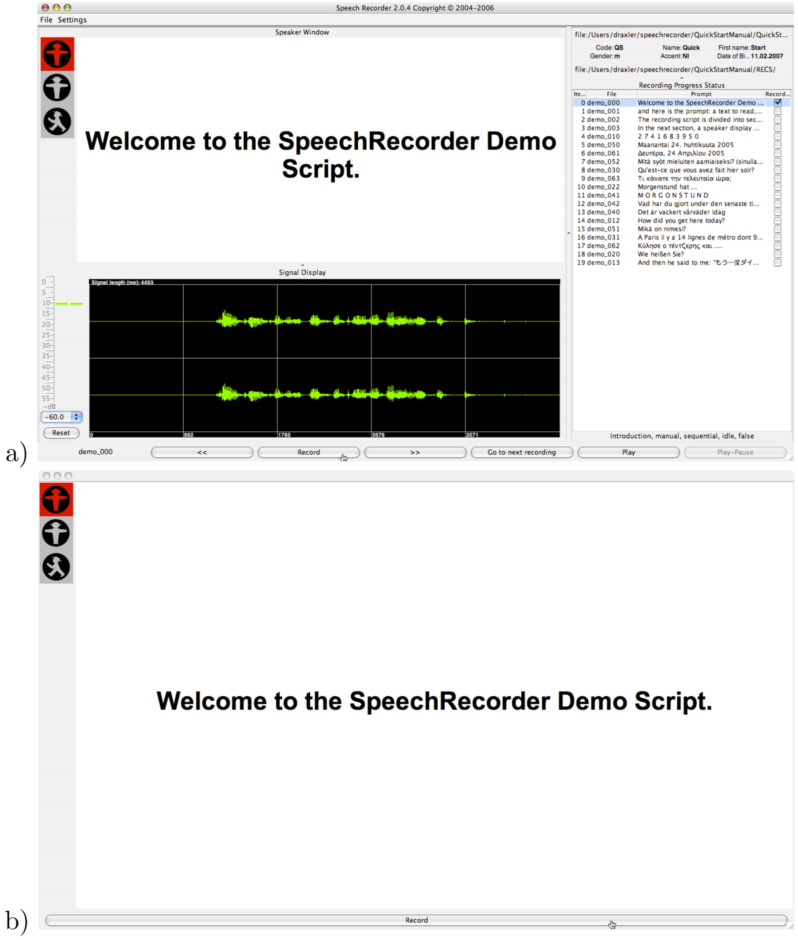
You're done – you've recorded your first session using *SpeechRecorder*!

Demo Script

The demo script consists of two sections. The first section contains test items, recording progress is manual (i.e. you have to click to start and end a recording, and click 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)).



Welcome to the SpeechRecorder Demo Script.

file:/Users/draxler/speechrecorder/QuickStartManual/QuickSt...

Code: **QS** Name: **Quick** First name: **Start**
Gender: **m** Accent: **Nl** Date of Bi... **11.02.2007**

file:/Users/draxler/speechrecorder/QuickStartManual/RECS/

Recording Progress Status

Ite...	File	Prompt	Record...
0	demo_000	Welcome to the SpeechRecorder Demo ...	☑
1	demo_001	and here is the prompt: a text to read,...	☐
2	demo_002	The recording script is divided into sec...	☐
3	demo_003	In the next section, a speaker display ...	☐
4	demo_010	2 7 4 1 6 8 3 9 5 0	☐
5	demo_050	Maanantai 24. huhtikuuta 2005	☐
6	demo_061	Δευτέρα, 24 Απριλίου 2005	☐
7	demo_052	Mitä syöt mieluiten aamiaiseksi? (sinulla...	☐
8	demo_030	Qu'est-ce que vous avez fait hier soir?	☐
9	demo_063	Τι κάνατε την τελευταία ώρα;	☐
10	demo_022	Morgenstund hat ...	☐
11	demo_041	M O R G O N S T U N D	☐
12	demo_042	Vad har du gjort under den senaste ti...	☐
13	demo_040	Det är vackert vårväder idag	☐
14	demo_012	How did you get here today?	☐
15	demo_051	Mikä on nimesi?	☐
16	demo_031	A Paris il y a 14 lignes de métro dont 9...	☐
17	demo_062	Κύλησε ο τέντζερης και	☐
18	demo_020	Wie heißen Sie?	☐
19	demo_013	And then he said to me: "もう一度ダイ...	☐

Introduction, manual, sequential, idle, false

Welcome to the SpeechRecorder Demo Script.

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

Contents

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* 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.

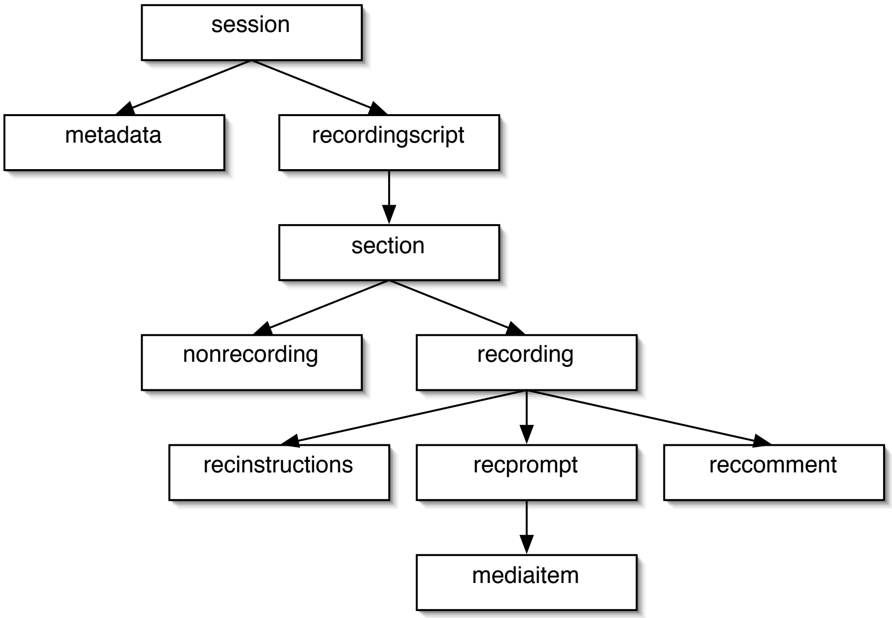


Figure 2: Structure of a *SpeechRecorder* recording script

A recording script is stored as an XML document. The DTD is given in Appendix A. *SpeechRecorder* does not yet have an editor for recording scripts. Hence, recording scripts must be created and edited using an external XML editor.

1.1 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. *autoprogress* 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 C.1 for problems when using audio or video prompts).

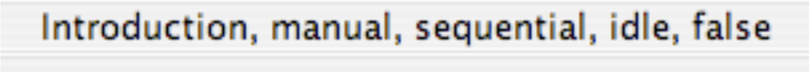
Sample sections

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

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

Section display

Information on the section is displayed below the table with the recording items in the supervisor view (fig. 3).



Introduction, manual, sequential, idle, false

Figure 3: Section information display in the supervisor view

1.2 The `<recording>` element

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

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

<!ATTLIST recording itemcode CDATA #REQUIRED
                    recduration CDATA #REQUIRED
                    prerecdelay CDATA #IMPLIED
                    postrecdelay CDATA #IMPLIED
                    finalsilence CDATA #IMPLIED
                    beep CDATA #IMPLIED
                    rectype 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 attributes `itemcode` and `recduration` of `<recprompt>` are mandatory. They uniquely identify a recording item and specify the duration of the recording of this 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).

`recduration` specifies the recording time in milliseconds. `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).

`finalsilence` is a flag for silence detection to stop recording. If it is set to a value > 0, recording stops after the specified amount of silence. `beep` is a flag that determines whether a beep is to be played prior to recording (see C for details).

Finally, `rectype` is one of *audio* or *video* (see C for details).

Recording sample

```
<recording prerecdelay="2000"
           recduration="20000"
           postrecdelay="500"
           itemcode="demo_001">
```

```
    <recprompt>
        ...
    </recprompt>
</recording>
```

1.3 The `<mediaitem>` element

The `<mediaitem>` element holds the prompt item. It may be empty, or contain text which is displayed on the screen.

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

<!ATTLIST mediaitem mimetype 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. The encoding of the prompt text is inherited from the encoding of the entire recording script and hence for text prompts, this attribute is not used. However, 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.

`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 or video 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 1.3.

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

`<mediaitem>`sample

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

mimetype	src	alt	autoplay	modal	width	height	volume
text/UTF-8	–	–	–	–	–	–	–
text/rtf	URL	–	–	–	–	–	–
image/jpeg	URL	+	–	–	+	+	–
audio/x-wave	URL	+	+	+	–	–	+
video/mpeg	URL	+	+	+	+	+	+

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

```
<mediaitem mimetype="text/UTF-8">
    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" />
```

This `<mediaitem>`element shows an image loaded from a 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 seqence of phases is shown in fig. 4.

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*.

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

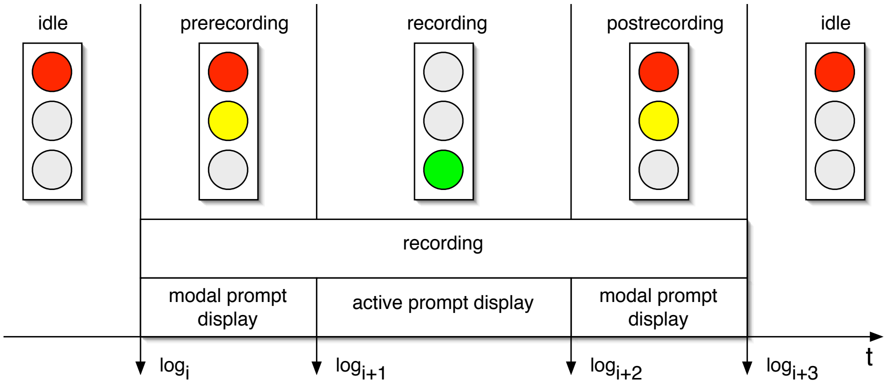


Figure 4: Recording phases

the utterance. A postrecording phase is most commonly used to avoid signal truncation due to clicking the `stop`button too early.

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`

The menu `File` contains commands to create, open, close, import, and save projects and to quit the application.

3.1 `New` command

The `New` command prompts the user for a project name and creates a project directory at the location provided. Initially, this directory contains an empty speaker database, a directory for the audio recordings, a project configuration file, a sample recording script, and the recording script DTD.

3.2 `Open` command

The `Open` command prompts the user to select a project from the list of known projects. 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.

3.3 `Close` command

`Close` closes the current project. If this project has been changed, e.g. by editing

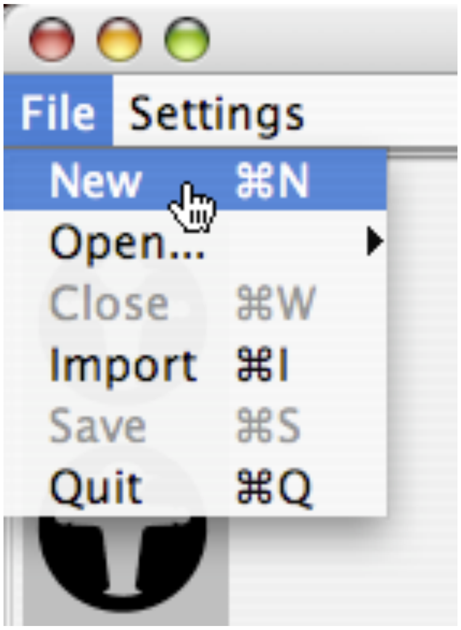


Figure 5: `File` menu

the speaker database, the user is prompted to save or discard the changes prior to closing the project.

3.4 `Import` command

The `Import` command for a project archive in a zip-archive. The archived project will be deployed in the project directory.

3.5 `Save` command

`Save` saves the current speaker database and project settings in local files in the project directory.

3.6 `Quit` command

`Quit` exits the application. The user is prompted to save any changes to the current project.

4 Menu `Settings`

The `Settings` menu allows the user to configure the current project, to edit the speaker database, to set the recording parameters, to skip to a given recording

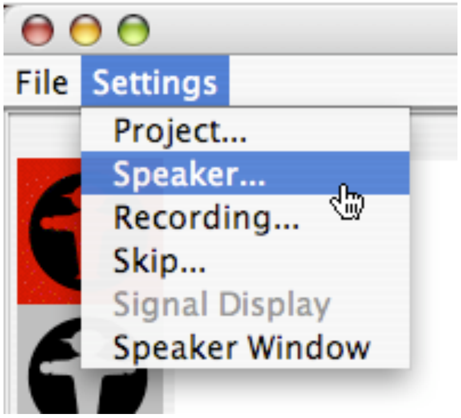


Figure 6: `Settings` menu

item, and to toggle the speaker display on and off.

4.1 `Project...` command

The `Project...` command opens a dialog window with the tabs `Project`,`Audio`, `Speakers`, `Recording`, and `Prompting`.

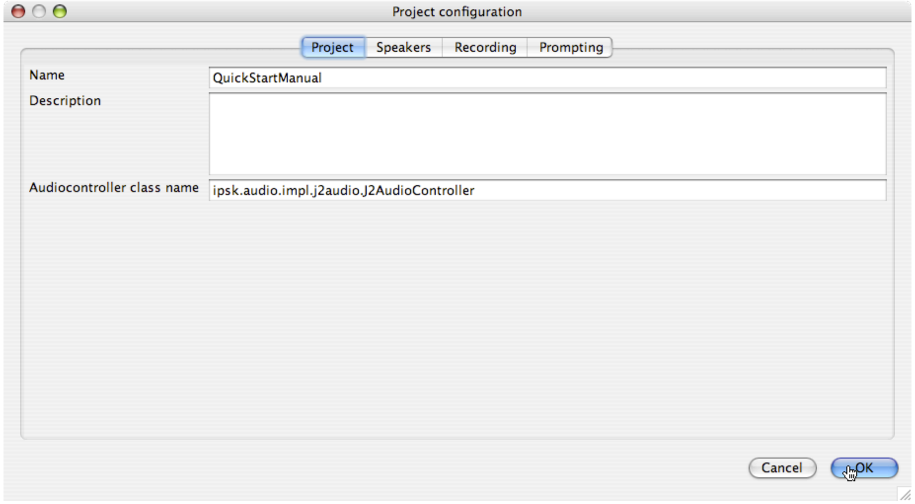
The `Project` tab (fig. 7) presents the project name, its location on disk, and the audio class used to record audio.

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 (fig. 8) allows the user to set the recording parameters. They include the sample rate, quantization, byte order, encoding, number of channels, whether repeated recordings of an item overwrite previous ones or are stored as versions, the progress mode, resetting the level meter for every recording, default values for pre- and postrecording phases, and the location for the recorded audio files.

Note that if the location for recorded audio files begins with `http://`, then the files are saved to a server via the `http` protocol over the Internet. In this case, the server must be configured to accept input via web forms with data transferred using the post method (see **??** for details).

The `Prompting` tab (fig. 9) displays the lists of fonts for prompt and instructions texts, and the location of the recording script file. This recording script file can be stored in the project directory or in any accessible location in the local file system.



Name: QuickStartManual

Description:

Audiocontroller class name: ipsk.audio.impl.j2audio.J2AudioController

Figure 7: `Settings > Project...` command

4.2 `Speaker...`command

`Speaker...` opens the speaker database and allows entering, deleting or selecting a speaker.

4.3 `Recording...` command

`Recording...` shows an audio mixer allowing the user to select input and output devices and their levels. Note that JavaSound does not detect all mixer controls of a given hardware configuration. If you do not see your input devices in the list, then select them via the system control panel of your operating system.

4.4 `Skip...` command

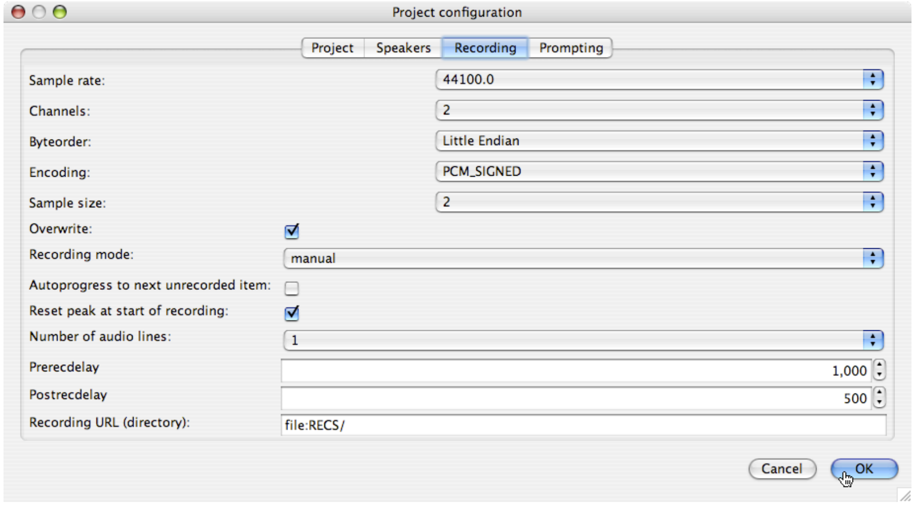
`Skip...` prompts the user for a script item number and skips to the appropriate item in the recording script.

4.5 `Signal display` command

Not yet implemented.

4.6 `Speaker Window` command

The `Speaker Window` command toggles the speaker view on the secondary displays on and off.



Sample rate: 44100.0
Channels: 2
Byteorder: Little Endian
Encoding: PCM_SIGNED
Sample size: 2
Overwrite: ☑
Recording mode: manual
Autoprogress to next unrecorded item: ☐
Reset peak at start of recording: ☑
Number of audio lines: 1
Prerecdelay 1,000
Postrecdelay 500
Recording URL (directory): file:RECS/

Figure 8: `Settings > Project... > Recording` dialog window

5 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.

To address a server via the Internet, the server address must be provided as a URL in the `Settings > Project > Recording` dialog:

```
http://SERVER_NAME/SERVER_PATH
```

with `http://`the data transfer protocol, `SERVER_NAME` the IP-name of your server, and `SERVER_PATH`the directory on the server. *SpeechRecorder* will then encode the data to be sent to the server as attribute-value pairs for the following attributes:

cmd command to server

store_audio audio signal data

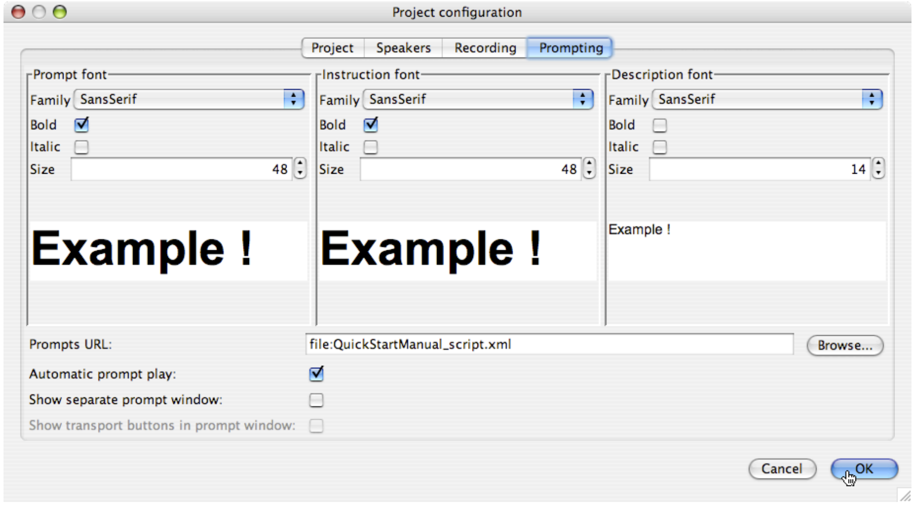
store_log log file data

store_timelog log file timestamp

itemcode unique id within a recording session

speakercode speaker code

speakerid unique speaker id



Prompt font
Family SansSerif
Bold ☑
Italic ☐
Size 48

Instruction font
Family SansSerif
Bold ☑
Italic ☐
Size 48

Description font
Family SansSerif
Bold ☐
Italic ☐
Size 14

Prompts URL: file:QuickStartManual_script.xml
Automatic prompt play: ☑
Show separate prompt window: ☐
Show transport buttons in prompt window: ☐

Figure 9: `Settings > Project...> Prompting` dialog window

extension signal filename extension

script name of recording script

session session id

line

recversion version of the recording file; augmented for every re-recording of the same prompt

Your server must correctly extract and interpret the attribute values and read the signal data which is sent via the post method. This is usually achieved by a cgi-module in the server, or external scripts called by the server, or a Java server such as Apache Tomcat.

6 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)>

<!ATTLIST session id CDATA #REQUIRED>


<!ELEMENT metadata (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 (mediaitem)>

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

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


<!ELEMENT recinstructions (#PCDATA) >

<!ATTLIST recinstructions    mimetype CDATA #IMPLIED
                            src CDATA #IMPLIED >


<!ELEMENT recprompt (mediaitem)>
```

```
<!ELEMENT reccomment (#PCDATA)>

<!ELEMENT mediaitem (#PCDATA)*>

<!ATTLIST mediaitem    mimetype 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, autoprogress, autorecording.*

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/utf-8* 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`.

- No recording script editor. Recording scripts have to be edited using an external XML editor.
- The following attributes are defined in the recording script DTD, but have not yet been implemented in *SpeechRecorder*:
  - The `<nonrecording>` element is not yet implemented.
  - `mimetype` and `src` attributes for the `<recinstructions>` element are ignored.
  - Playing a beep before and silence detection to stop a recording and are not yet implemented.

  - Recording 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`, or `0-9` for the `itemcode` attribute.

C.1 Time-dependent prompts

The following list describes issues with time-dependent prompts, i.e. audio and video prompts. Note that the playback of the time-dependent prompt will be recorded.

- For time-dependent prompts set the `promptphase` attribute of the enclosing section to *recording*. Otherwise, multiple playbacks of the time-dependent prompt may occur.

  Furthermore, make sure that the recording duration is longer than the duration of the time-dependent prompt. Otherwise, the playback of the prompt will continue until after the end of the recording duration – it may even overlap with the playback of the subsequent time-dependent prompt.

C.2 Platform dependencies

- *Mac OS X:*Audio device selection does not work in Mac OS X and Java versions prior to Java 2 1.5.0_07. The sample rate must be set to 44.100 kHz, 16 bit stereo, PCM in the `Settings` dialog, and only one audio device may be connected. Note: on Mac OS X notebooks, this audio device is already used by the internal microphone.

- *Windows XP:* If a Windows beep is output via an M-Audio mobile pre USB device, the recording sample rate is reset to the sample rate of the beep, i.e. usually 22.050 kHz. *SpeechRecorder*does not detect this change of sample rate. All subsequent recordings will be made using the new sample rate.

D 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.

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.