



INSTITUTE FOR INFORMATION TECHNOLOGY  
AND COMMUNICATIONS (IIKT)

# Annotation design manual

ikannotate2

Version 01.11.2018

Ingo SIEGERT



## CHAPTER 1

# Installation

---

Until now, ikannotate2 does not come with an install-routine. Therefore, just unpack the delivered Zip-archive into a folder of your choice.

The program is developed and tested under Windows 7 and Windows 8. The folder of the windows version contains the executable (ikannotate2.exe) and all needed DLLs. As well as the Microsoft Visual C++ 2005 Redistributable Package (x64) (vcredist\_x64.exe), which is needed to run ikannotate2 and installs the following two DLLs:

- `msvcp120.dll`
- `msvcr120.dll`

For the Linux Version, you need to install the latest stable QT5-libraries, at least the following packages:

- `qt5-multimedia`
- `qt5-base`
- `qt5-gstreamer`

Furthermore the folder should contain the following files:

- `ikannotate.lic` – your personal licence file
- `Labels.conf` – a text-file to define the list-based labels
- `ikannotate.conf` – the configuration file

and the following folder(s) {Windows Version only}:

- `Data` - Folder to place the data to be annotated
- `{audio}` QT5 specific folder for audio playback
- `{mediaservice}` QT5 specific folder for video playback
- `{platforms}` QT5 specific folder for GUI-Display



## CHAPTER 2

# Preparation of Annotation

---

The annotation can be conducted with three different methods:

1. free-definable (Emotion) Word Lists,
2. an Emotion Wheel and
3. a non-verbal Pictorial Annotation Method for Pleasure-Arousal-Dominance

The program aims to support the annotation process, by making it as easy as possible. The annotator is given a very easy to understand and intuitive interface, where he can focus completely on the annotation. Furthermore, the program restricts the options of the annotator, he has to listen to the data in the pre-defined order, is only allowed to set the label after listening to the actual sample (although he can listen to the sample several times), and only after giving an assessment, the next file can be annotated. Also the progress is saved automatically after each assessment, so that the annotator can interrupt the annotation at any time, without the need to remember to save the actual work.

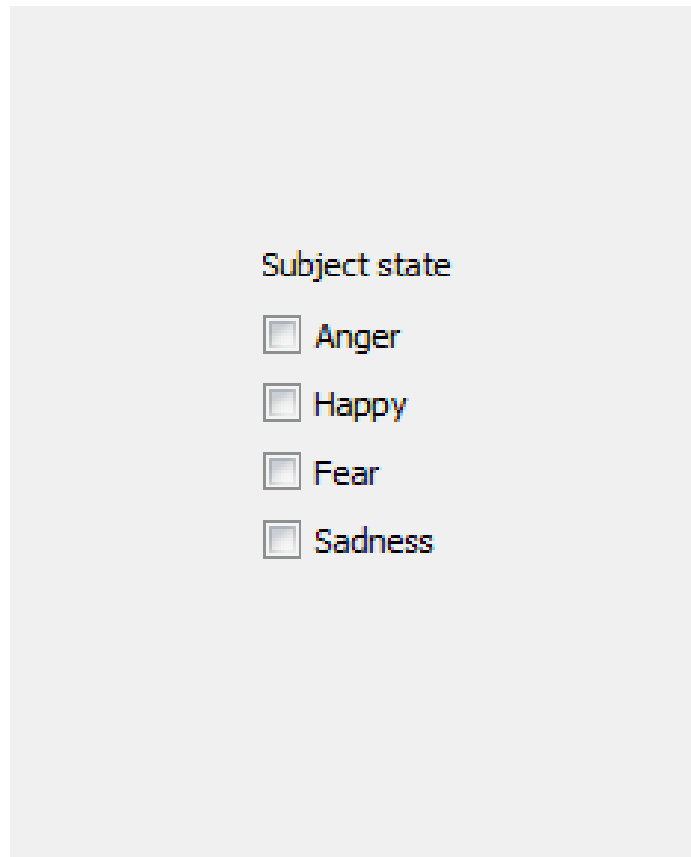
But on the other hand the designer or developer of the annotation has a lot of options to decide on the method. He can chose on the annotation method, and the order of samples. Furthermore, he can specify the emotion terms for the Emotional Word Lists and if for the Emotion Wheel the annotation should be conducted in a single step or via a step-wise process.

The desired method can specified in the [Labelling]-Part of `ikannotate.conf`:

Emotion Word Lists:	Pleasure-Arousal-Dominance:	Emotion Wheel:
<code>LabellingType=EWL</code>	<code>LabellingType=PAD</code>	<code>LabellingType=Wheel</code>

## 2.1 Emotion Word Lists

This method is the easiest one, the annotator can select from a list of predefined emotion terms, which have to be specified in advance by the study-designer. The Word-List is presented as follows within ikannotate2



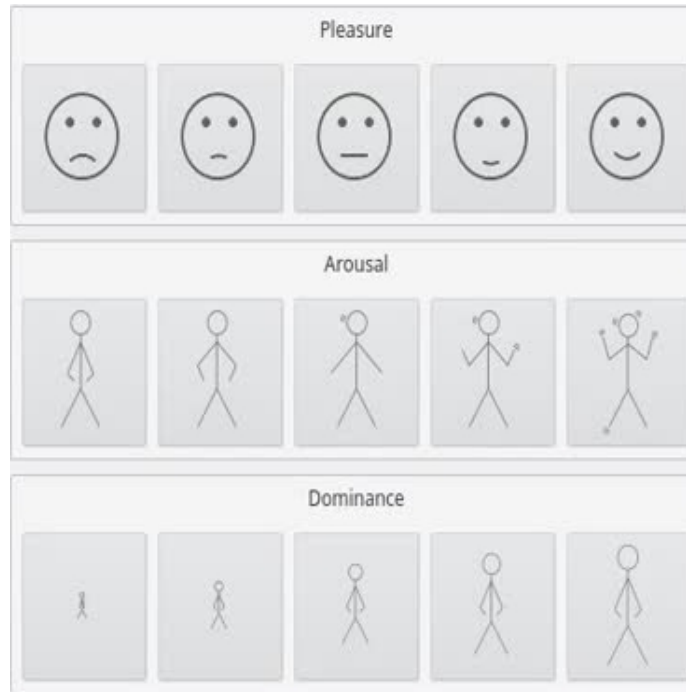
**Figure 2.1:** Example of Emotion Word List

Additionally for Emotion Word Lists the desired terms have to be specified in the file `Labels.conf` using the following Format:

```
Title=Subject state  
Label=Anger|Happy|Fear|Sadness
```

## 2.2 Pleasure-Arousal-Dominance

This is a non-verbal method, to assess affective reactions along the dimensions of Pleasure, Arousal and Dominance. It consists of three rows of icons which respectively detect a dimension on a five-point scale by stylized figures. The method is widely used because it is acquitted and carried out very quickly.



**Figure 2.2:** Non-verbal pleasure arousal dominance labelling method

This method is based on the Self-Assessment Manikins by: M. Bradley and P. Lang, “Measuring emotion: the Self-Assessment Manikin and the Semantic Differential.” *J Behav Ther Exp Psychiatry*. 1994 Mar;25(1):49-59.

The dimension **Pleasure** describes the positive or negative feeling caused by a situation, an object or an event. E.g. anger and anxiety are supposed to be unpleasant, joy is supposed to be pleasant.

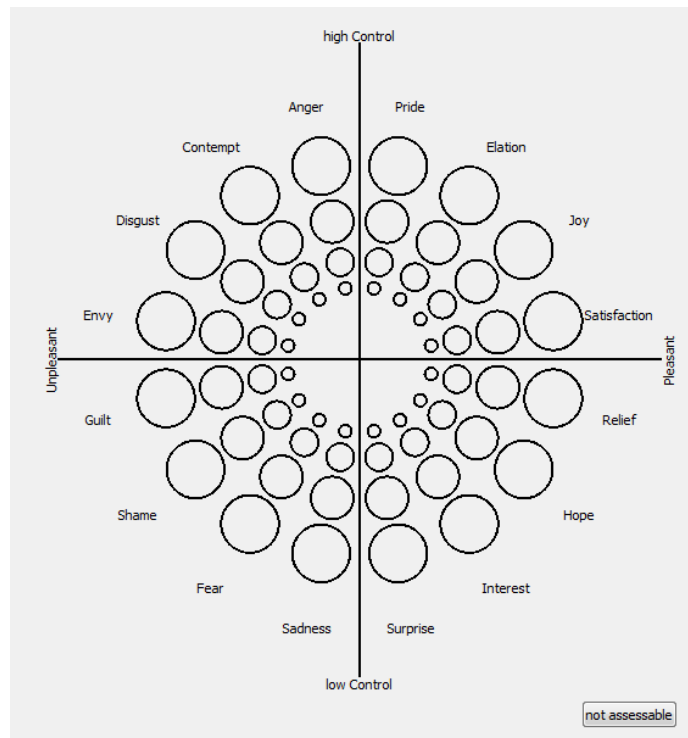
The dimension **Arousal** describes the perceived vigilance as a physiological and psychological condition of a person. The range reaches from excitation to dozziness or boredom.

The dimension **Dominance** describes how much a person feels in control of a situation. A small manikin means that the proband feels he/she has no power to handle the situation.

## 2.3 Emotion Wheel

This method is a theoretically derived and empirically tested instrument to measure emotional reactions to objects, events, and situations. The respondent is asked to indicate the emotion he/she experienced by choosing intensities for a single emotion or a blend of several emotions out of 20 distinct emotion families. The emotion families are arranged in a wheel shape with the axes being defined by two major dimensions of emotional experience. This method is based on the Geneva Emotion Wheel:

Scherer, K. R. (2005). What are emotions? And how can they be measured? *Social Science Information*, 44(4), 693-727.



**Figure 2.3:** Emotion Wheel, full wheel displayed at once.

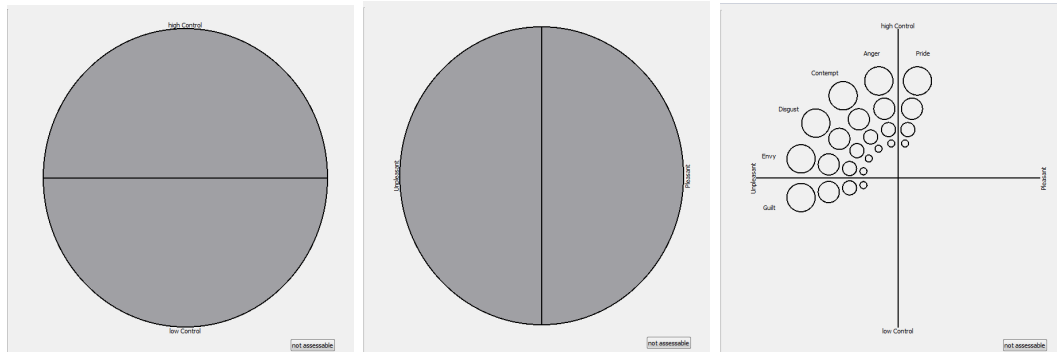
ikannotate2 supports two different labelling approaches for this method. Either the full wheel with all 16 emotions is displayed at once and the annotator can choose between all terms in one step. Or to simplify the annotation, a step-wise approach is offered: a) The annotator decides whether high or low dominance exists. b) He decides between pleasant and unpleasant feelings. c) The resulting quadrant is displayed and the annotator selects one of the emotions and its intensity. The desired method can



specified in the `ikannotate.conf`:

Method 1: `stepwise=false`

Method 2: `stepwise=true`



**Figure 2.4:** Emotion Wheel, stepwise labelling approach.

## 2.4 Data Preparation

To be able to annotate data with `ikannotate2`, the desired Method has to be specified in the File `ikannotate.conf`, for Emotion Word Lists the needed labels have to be specified in the File `Labels.conf` as well.

Afterwards, the Data can be prepared. Actually `ikannotate2` is designed to process single audio or video files. Therefore, the samples to be analysed has to be segmented manually. Afterwards, these segments have to be placed in the folder `Data` as a sub-directory of the `ikannotate2` binary, e.g.:

Data

–sample1.wav

–sample2.wav

–sample3.wav

...

Furthermore, a set-file (`*.data`) has to be generated that contains a list all needed sample files (one per line) and the filename for the result-file. The path to the sample data has to be relative, starting from the `Data`-Folder of `ikannotate2`. A sample `Set.data` could be look like:

```
resultFilename:FILENAME  
sample1.wav  
sample2.wav  
sample3.wav
```

It is possible to create several sets at once. Also sub-folders within the **Data-Folder** are possible. The order, `ikannotate2` is using to present the samples to the annotator is set by the order of the files within the `*.data-File`.

`ikannotate2` is able to play all multimedia-files that can be natively played under Windows as well. `ikannotate2` uses the same codecs as Windows, if you are unsure if a specific Format can be played, test it in advance using the Windows MediaPlayer.

If everything is tested, the program can be used for annotation.

### 2.4.1 Result-File

The Results are written line-by-line into the in the data-file specified file (`resultFilename:XXX`. The result file contains the names of the annotated items together with their indicated labels, separated by “;”:

```
sample1.wav; Anger  
sample2.wav; Fear  
sample3.wav; Joy
```

### 2.4.2 EXMARALDA

`ikannotate2` is prepared to work with EXMARALDA partitur XML-Files. Therefore, you can load the XML-file into `ikannotate2`. The programm will ask you, if a existing tier should be used for the labels or a new one should be generated. The generation of a `Set.data` file is not necessary.

Be aware that this method is not well tested and might be buggy. (Bubgs can be reported and will be fixed.)

## CHAPTER 3

# Configuration Parameters

---

The configuration File `ikannotate.conf` has additional parameters that could be helpful for your annotation task. The following Listing prints all of them together with their definitions.

```
[Main]
MeasureLabellingTime=[bool]
doAutomaticResultUpload=[bool]
resultCollectServer=[URI/collect.php]

[Labelling]
labellingType=[EWL|Wheel|PAD]
stepwise=[bool]
Uncertainty=[bool]
UncertaintyMin=[int , default=0]
UncertaintyMax=[int , default=5]

[Questionnaires]
usePostQuestionnaires=[bool]
Likertsize=[int , default=5]
Questionnairefile=[String , default=Questions.txt]
```

### 3.1 [Main]

#### 3.1.1 MeasureLabellingTime

This option can be used to measure the time an annotator needs to decide for a label. The time measurement starts when the audio/video-file is started for the first time and stopped when the annotator presses the next button. (If Uncertainty is activated, there is a second time point measurement given after Uncertainty labelling is finished (total time)). To know if the audio/video-file has been played several times,

an additional File-Counter is given. The measurement is done in milliseconds and appended to the result file as follows:

```
sample1.wav; Anger; 1000; 2000; 1  
sample2.wav; Fear; 1000; 2000; 1  
sample3.wav; Joy; 1000; 2000; 1
```

### 3.1.2 doAutomaticResultUpload & resultCollectServer

To facilitate the external use of ikannotate2, where the annotators can use the tool on their own devices at home, ikannotate2 offers the possibility to automatically upload the results onto a webserver.

Therefore, the option `doAutomaticResultUpload` has to be enabled and a `resultCollectServer` including the PHP-Script has to be specified. The Server has to be in the position to being able to store data delivered via the GET-routine.

If this option is enabled, ikannotate2 asks for an identifier (preferable the annotator's name) at the end of the data-set and sends this identifier together with the resultfile content to the PHP-script of the Server using the following format:

```
resultCollectServer?filename=%1&data=%2
```

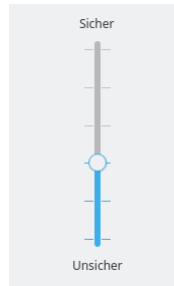
The PHP-Script has to output a document containing either OK or ERROR, to indicate the file processing.

## 3.2 [Labelling]

### 3.2.1 Uncertainty & UncertaintyMin, UncertaintyMax

To also get insights how confident the annotators are with their labels, the designer can add the Uncertainty Option. Hereby ikannotate2 ask the annotator after every item about their Uncertainty with his choice. This inquiry is implemented with a slider:

Using the Options `UncertaintyMin` and `UncertaintyMax`, the value range of the slider can be changed.



**Figure 3.1:** Uncertainty-slider

The Uncertainty is stored in the results-file as well. Note! higher numbers denote a higher certainty!:

```
sample1 .wav; Anger; 1
sample2 .wav; Fear; 4
sample3 .wav; Joy; 5
```

If `Uncertainty` and `MeasureLabellingTime` are enabled, the result-file will first denote the uncertainty and afterwards the uncertainty:

```
sample1 .wav; Anger; 1; 1000; 2000; 1
sample2 .wav; Fear; 4; 1000; 2000; 1
sample3 .wav; Joy; 5; 1000; 2000; 1
```

### 3.3 [Questionnaires]

ikannotate2 allows it to perform post questionnaires. Therefore at least the following configuration parameter has to be set: `usePostQuestionnaires=true`. An example of the corresponding implementation is depicted in Figure 3.2.

An Introduction text (Introduction) and two types of Questions are available:  
a) Free-form text and b) Likert-Scales.

Be aware that at the moment post-questionnaires are only supported for utterance-based annotations and not for EXMARALDA-Transkripts.

### 3.3.1 usePostQuestionnaires

This configuration enables the post-questionnaires. They are shown to the annotator after the set of utterances is done.

### 3.3.2 Likertsize

This parameter changes the size of the Likert-scale. As default-value 5 is chosen.

### 3.3.3 Likertsize

This parameter changes the size of the Likert-scale. As default-value 5 is chosen.

### 3.3.4 Questionnairefile

When the parameter `usePostQuestionnaires` is set, `ikannotate2` expects to read the Questions from the file `Questions.txt`. Otherwise use the parameter `Questionnairefile` to specify the actual file.

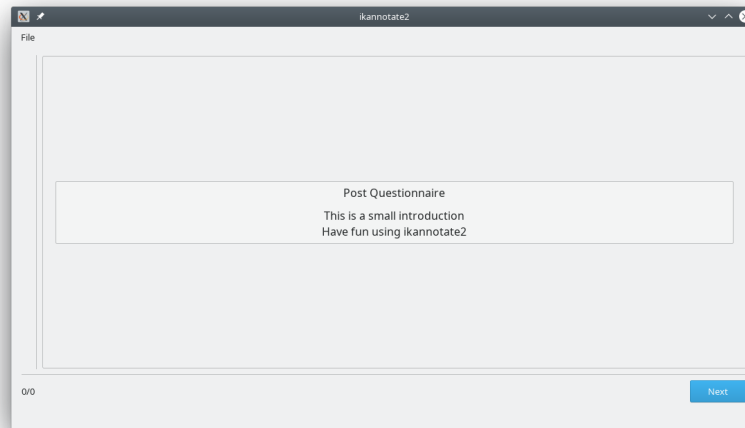
The format of the file is as follows:

```
Questiontype | Title##Question
```

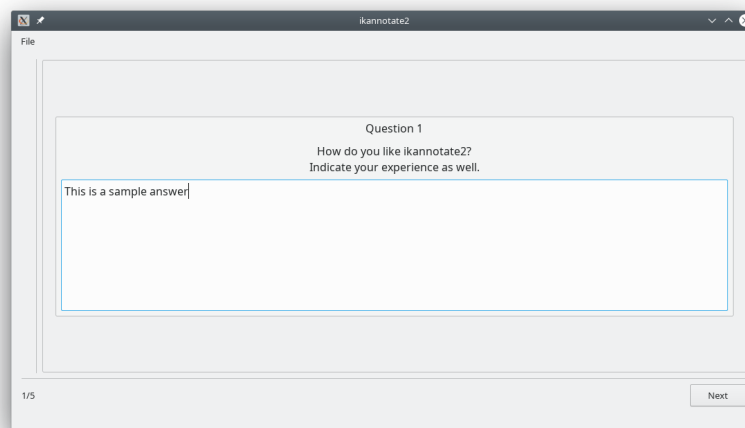
The `Questiontype` can be either `Introduction`, `Freeform`, or `Likert`. The title is separated by the vertical bar (“|”). This free-choosable string is displayed as headline for each question. Be aware, that `ikannotate2` does not support line-breaks within titles, thus the length should be checked beforehand.

The question text is then distinguished by the double hash delimiter (“##”). Again a free text can be used and line-breaks are supported for questions, but have to be indicated manually by using `<br>`.

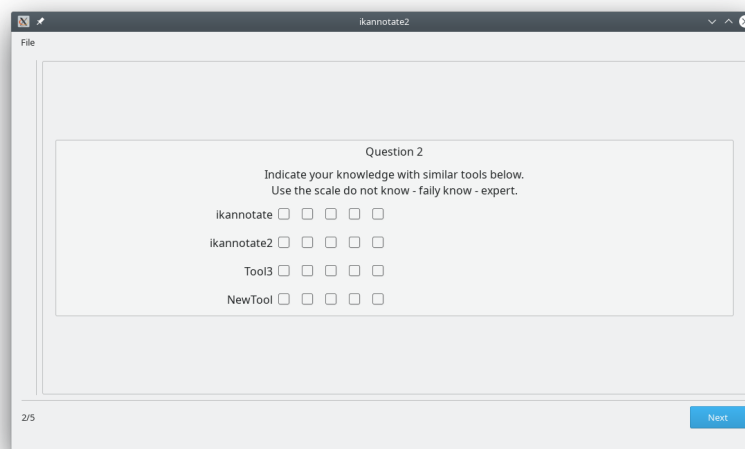
For Likert-type Questions the initial question and the categories have to be separated. Therefore again the double-hash (“##”) is used as delimiter. Furthermore, a single hash (“#”) is used as delimiter between the categories itself. `ikannotate2` automatically generates as many category entries as needed. For each entry the checkboxes are connected so that only one value can be chosen per category.



(a) Post-questionnaire “Introduction”



(b) Post-questionnaire “Freeform”



(c) Post-questionnaire “Likert”

**Figure 3.2:** Examples for the post-questionnaire implementation of ikannotate2





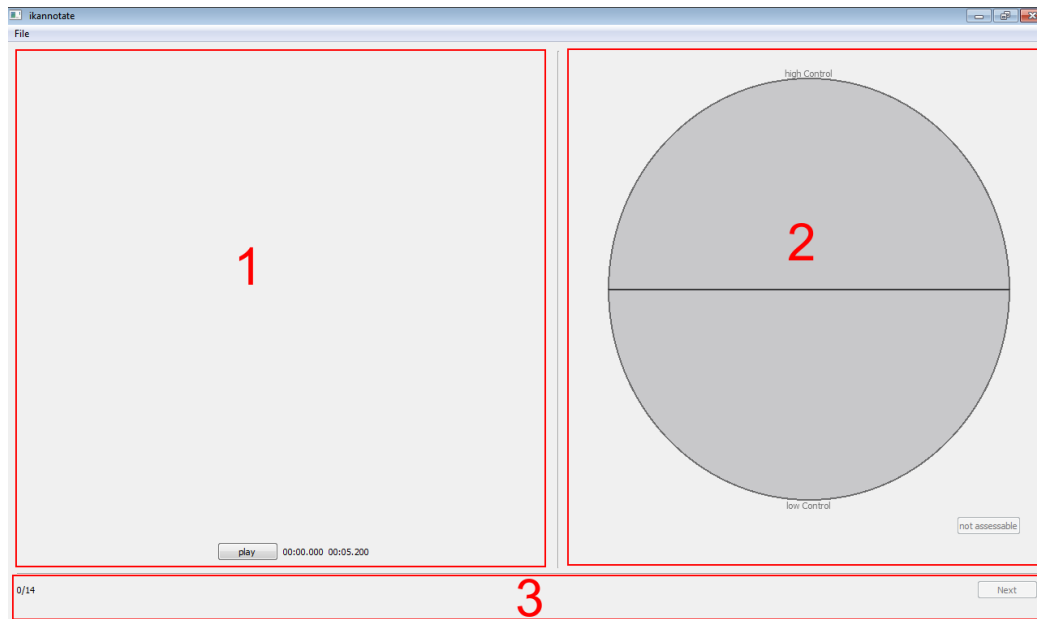
## CHAPTER 4

# Annotation Usage

---

The annotator just have to start the program and load the desired Set and afterwards can start the annotation. To load a set, the annotator has to open the File-Dialog by **File**→**read set**. Afterwards, the set is loaded and three different parts can be distinguished:

1. Player-Part
2. Annotation-Part
3. Control-Part



## 4.1 Player-Part

The Player-Part shows the actual sample. It shows actual time and end-time of the sample. By clicking on the Play-Button the sample is played-back. The annotation-Part is disabled as long as the file hasn't been played completely once.

## 4.2 Annotation-Part

The annotation part hosts the chosen annotation method, by default it is disabled for the actual sample. After the sample is played-back at least once. When the annotator has assessed the actual sample, the Control-Part is enabled and the next sample can be assessed.

## 4.3 Control-Part

The Control-Part has on the left side a counter of the size of the actual Set<sup>1</sup>. On the left side the Button Next is displayed. As long as the user has not assessed the actual sample. If the user presses the Next button, the actual result is stored automatically in the specified result file and the assessed sample is removed from the loaded Set-File. Thus at the end of the set, the Set-File is empty and the result-File contains all assessments. Where each line hold the filename of the sample and separated by “;” the assessed emotion. The resultfile for the above given sample Set-File would look like:

```
sample1.wav; Anger  
sample2.wav; Fear  
sample3.wav; Joy
```

---

<sup>1</sup>Be aware that due to ikannotate2s processing, the counter is decreased every time the set is restarted, as ikannotate2's auto-save function removes already assessed samples from the data-set file

## APPENDIX A

# Example Scripts

---

### A.1 PHP-script for Result Storage

```
1 <?php
2 $Filename = $_GET['filename'];
3 $Data = $_GET['data'];

6 $data_array = explode("|", $Data);

8 // write results into file:
9 $myfile = fopen("$Filename.results", "w") or die("Unable to open file!");
10     foreach($data_array as $value){
11         fwrite($myfile, $value.PHP_EOL);
12     }
13 fclose($myfile);

15 // read written results for test:
16 $myfile = fopen("$Filename.results", "r") or die("Unable to open file!");
17 $writtendata = fread($myfile, filesize("$Filename.results"));
18 fclose($myfile);

20 // replace linebreaks by |
21 $linebreaks = array("\n", "\r\n");
22 $writtendata = str_replace($linebreaks, "|", $writtendata);
23 $writtendata = str_replace("||", "|", $writtendata);

25 // compare results:
26 if($writtendata == $Data){
27     echo "OK";
28 } else {
29     echo "ERROR";
30 }

32 ?>
```