# First steps to create an experiment in DMDX/OpenSesame/E-prime: prepare the audiofiles with Praat

******

You have successfully prepared stimuli, they are all recorded, now what?

First, you need to cut every word/item/stimulus and save it as one audio file (.wav).
If you have 300 words or more, that can be a daunting task. There are ways to automatize this, but let me talk here about doing this with Praat and a Praat script.

The first steps at this point are

**1) create an excel file with the filenames that you will need in the scripts** to use with your experiment presentation software (for example, E-Prime, DMDX or OpenSesame software).
The excel file should contain the filenames with an explanation of the different conditions with which they will be grouped. Most likely, you would take the excel sheet you used to create your stimuli and add the audio-file codes to it.
Below is an example. The audiofile coding is "transparent" and short. Most importantly, it has no spaces (even when the suffix <_s> is added at the end to specify that the word was recorded in sentence context).

| | | **Condition** | | | **audiofiles (_s = sentence context)** | |
|---|---|---|---|---|---|---|
| **Word** | **English gloss** | **place** | **aspiration** | **voicing** | **coding** | |
| caca | be clear | dental | unasp | voiceless | DUVL1 | |
| caca | be clear | dental | unasp | voiceless | DUVL1 | _s |
| chaya | spread out | dental | asp | voiceless | DAVL1 | |
| chaya | spread out | dental | asp | voiceless | DAVL1 | _s |
| gcaba | make a incision | dental | unasp | voiced | DUVD1 | |
| gcaba | make a incision | dental | unasp | voiced | DUVD1 | _s |
| ngcama | feast | dental | nasal | voiced | DNVD1 | |
| ngcama | feast | dental | nasal | voiced | DNVD1 | _s |
| qala | start | postalveolar | unasp | voiceless | PUVL1 | |
| qala | start | postalveolar | unasp | voiceless | PUVL1 | _s |
| gqaba | paint (face) | postalveolar | unasp | voiced | PUVD1 | |
| gqaba | paint (face) | postalveolar | unasp | voiced | PUVD1 | _s |
| ncama | give up | dental | nasal | voiceless | DNVL1 | |
| ncama | give up | dental | nasal | voiceless | DNVL1 | _s |

These are the codings you would use in the TextGrid. This will also be the name of the audiofile. Having such an excel file allows you later to quickly see what file belongs to which condition

**2) Mark each stimulus/word/item that you want to cut into its own file** and give its specific name. This involves (with Praat) creating a Textgrid for your sound file, and start putting in the boundaries for each stimulus/word/item.

Before you start working, create a specific folder "Praat" or "Stimuli" for example, where you **make a copy** of your big sound file that you'll splice (just in case something happens). It is best

to work directly from the Desktop and to use folder and file names WITHOUT any spaces. Praat does not like spaces in filenames. I commonly do this in a folder that I called "Praat" on my desktop. This makes the paths also much shorter in the scripts (see below), and less prone to typos.
Prefer something short (e.g. "`saakuru.wav`" instead of "`stimuli for abx.wav`") for all your wavfiles.

**Ready?**
You'll need the Praat Script `save-small-files.praat`
And your excel file where you have assigned the different filenames to each stimulus.

1. Open the long sound file (e.g., Chisato/saakuru.wav) using:
Read | Open long sound file

2. Create a TextGrid file with a single tier called "words"
Annotate | To TextGrid...

Tier names: **words**
Point tiers: <delete contents and leave blank>

{I use "words" as the name for that tier because I use that too in the Praat script used below to cut up the files. It's possible to use something different, but then it's necessary to change that in the Praat script too}

3. Place boundaries around the stimuli in the TextGrid, and label intervals in tier (by typing in the file name).
These labels will become the names of the small sound files.
Hint: you can click in the waveform or spectrogram, then press "Enter" to place the boundary (rather than clicking the little circle with the mouse). You can also select the word and press "Enter". This will set both start and end boundaries.
**Place your boundaries at zero crossings to avoid hearing clicks in the files later (see Praat instructions to know how to mark zero crossings).** Make sure not to place the end-boundary before the release of a consonant, for instance. Also make sure not to leave unequal silence portions before and after the word. What that means is: When placing the first boundary for a word, place it just before the beginning of the word, around 50 ms before. After the word, when you place the second boundary, make sure it's not 400 or 700 ms. after then end of the word, but again, rather close to it, 50 ms. This prevents that this silence in the sound files adds up unsystematically to the Inter=Stimulus=Intervals you have specified once you embed these sound files in your script.

4. Save the TextGrid file.

Make sure to double-check that all stimuli intervals have the right file name label, and that these filenames do not contain typos or spaces! It is important to be concentrated while doing this so as to not make mistakes. If you notice a mistake later, you will have to correct the TextGrid and re-run the script, which is not such a big deal, but it's best to do it right the first time.

5. Open save-small-files.praat:

Praat | Open Praat script...

Change the various parameters (name of long sound file, left and right buffers, prefix and suffix, etc.) if necessary. The "append time" option adds the start time of the small sound file (with respect to the long sound file) to the file name. This prevents overwrite in case there is more than one of the same label (producing, for example, lass-5.05.wav and lass-7.84.wav). We will not usually need this for our purposes but it is in the script in case you need it.
*****
Here a description of how to use the Praat script:
**save-small-files-README.praat**
**Praat-script_Savesmallfiles.pdf** is a commented script with explanations on how to modify the parameters for your own files.
******

N.B.: The Long sound object and the TextGrid object do not need to be
in the objects window when the script is run, but they can be.

6. Run the script:

Run | Run

DONE!! Congratulations!
You now have your stimuli ready to be implemented into the experiment! (But make sure to double check for mis-cuts, other issues like clicks in the files and you might need to normalize amplitude too if you have different voices).