

Export old LA questions

 altissia-launcher will only work if it's located at `/home/altissiadev/sources/altissia-launcher`.

An easy workaround is to create a symbolic link at this location and pointing to the real location in your own home directory.

This page documents the process to export and publish old LA questions.

Order of process

To achieve the export, you must follow those simple steps :

1. Convert the sounds from .wav to .mp3 (if needed)
2. Copy the sounds into the correct folder in altissia-launcher
3. Copy the questions (.xls files) into the correct folder in altissia-launcher
4. Launch the correct validation script
5. Check the validation output
6. Configure and launch the correct publication script
7. Check the publication output
8. Rename and put the update zip on the NAS
9. Tell Greg and J-P that the export is done and ready to publish

Convert the sounds from .wav to .mp3

If you need to publish new sounds, you will find them in `/data/newSources/leveltest/assets/sounds/` on the NAS.

As discussed with the linguistic team, the new sounds must be in their language folder in a version folder (ex: `/data/newSources/leveltest/assets/sounds/de/v20.3`). At the moment, it's not the case in every language but it will be corrected as we publish new sounds.

So, you go to the right folder : ex: `cd /data/newSources/leveltest/assets/sounds/de/v20.3/`

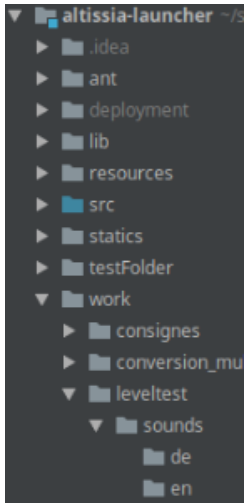
To convert the .wav files to .mp3, you will use a script from altissia-launcher. To use it, just do : `sh /home/altissiadev/sources/altissia-launcher/resources/scripts/sound-processing/wav-to-mp3.sh` (your path can be different, just adapt it)

Note: before using the script, check that [all the needed software is installed](#).

The conversion can take several minutes depending of the number of sounds to convert.

Copy the sounds into the correct folder in altissia-launcher

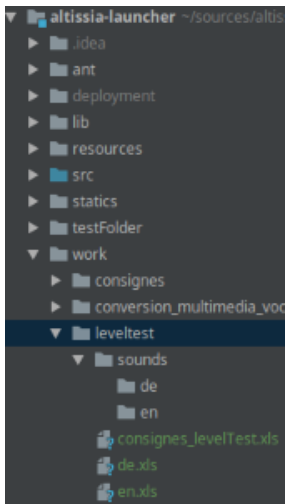
First, you need to **create a directory per language** for the sounds in `altissia-launcher/tools/work/leveltest/sounds/`. Each directory must be named as the language in lowercase.



Then **copy** the **.mp3** files in the right directory.

Copy the questions (.xls files) into the correct folder in altissia-launcher

Easy step, just copy the .xls files needed stored in /data/newSources/leveltest/ to altissia-launcher/tools/work/leveltest/ (don't forget to **copy the files for each language to update AND the instruction file leveltest_consignes.xls**)



Launch the correct validation script

Now that we have our .xls files and sounds (not mandatory) in the right place, let's configure the validation scripts.

There are two scripts in the file **altissia-launcher/statics/build.xml**:

- `leveltest.validate.question.excel`
- `leveltest.validate.question.language.C.excel`

The first one, is used to validate A and B languages. The other, for languages C and D.

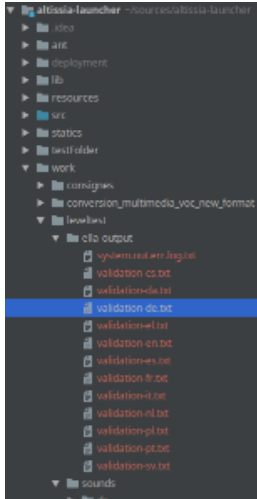
To use them, simply **click on the right script name** in your IntelliJ ant build.

This will generate a folder "ella-output" at **altissia-launcher/tools/work/leveltest**

Check the validation output

We have now a validation file for each language that the script process

ex : for leveltest.validate.question.excel :



You just need to **check the languages** you need to update. If there are errors, **send a mail to the linguistic team to tell them to correct the .xls files**.

!! Warning are not errors!

When everything is ok, we can use the publication script.

Configure and launch the correct publication script

Now that the validation is done, let's configure the publication scripts.

There are two scripts :

- `leveltest.publish.question.excel`
- `leveltest.publish.question.lang.c.excel`

The first one, is used to validate A and B languages. The other, for languages C and D.

The scripts should look like this :

This scripts should look like this

```
<macrodef name="leveltest.publish.question.excel">
    <attribute name="lg"/>
    <attribute name="lgUP"/>
    <sequential>
        <runLauncher applyOn="${work.dir}/leveltest/@{lg}.
xls" failonerror="false"
                                scriptName="leveltest.
publish.question.excel" preset="default"
                                params="
consignesPath=${work.dir}/leveltest/consignes_levelTest.xls" />
        <copy file="${work.dir}/leveltest/ella-output
/system.out.err.log.txt"
                                tofile="${work.dir}/leveltest/ella-output
/publication-@{lg}.txt"/>
        <move file="${work.dir}/leveltest/ella-output/@
{lg}"
                                tofile="${work.dir}/leveltest/ella-output
/data@{lgUP}"/>
        <mkdir dir="${work.dir}/leveltest/ella-output/data@
{lgUP}/sounds"/>
        <copy todir="${work.dir}/leveltest/ella-output
/data@{lgUP}/sounds">
                                <fileset dir="${work.dir}/leveltest/sounds
/sounds-@{lgUP}"/>
                                </copy>
        </sequential>
    </macrodef>

    <target name="leveltest.publish.question.excel">
        <leveltest.publish.question.excel lg="en" lgUP="EN"/>
        <leveltest.publish.question.excel lg="fr" lgUP="FR"/>
        <leveltest.publish.question.excel lg="de" lgUP="DE"/>
        <leveltest.publish.question.excel lg="es" lgUP="ES"/>
        <leveltest.publish.question.excel lg="it" lgUP="IT"/>
        <leveltest.publish.question.excel lg="nl" lgUP="NL"/>

        <leveltest.publish.question.excel lg="cs" lgUP="CS"/>
        <leveltest.publish.question.excel lg="da" lgUP="DA"/>
        <leveltest.publish.question.excel lg="el" lgUP="EL"/>
        <leveltest.publish.question.excel lg="pl" lgUP="PL"/>
        <leveltest.publish.question.excel lg="pt" lgUP="PT"/>
        <leveltest.publish.question.excel lg="sv" lgUP="SV"/>
        <zip file="${work.dir}/leveltest/update.zip"
basedir="${work.dir}/leveltest/ella-output">
            </zip>
    </target>
```

If you don't have sound to publish, just **comment those lines** :

```

<mkdir dir="${work.dir}/leveltest/ella-output/data@{lgUP}/sounds"/>
      <copy todir="${work.dir}/leveltest/ella-output
/data@{lgUP}/sounds">
                                <fileset dir="${work.dir}/leveltest/sounds
/sounds-@{lgUP}" />
                                </copy>

```

And comment the lines with the languages you don't need to publish. ex:

```

<leveltest.publish.question.excel lg="en" lgUP="EN"/>
<!--leveltest.publish.question.excel lg="fr" lgUP="FR"/-->
<leveltest.publish.question.excel lg="de" lgUP="DE"/>
<!--leveltest.publish.question.excel lg="es" lgUP="ES"/>
<leveltest.publish.question.excel lg="it" lgUP="IT"/>
<leveltest.publish.question.excel lg="nl" lgUP="NL"/>

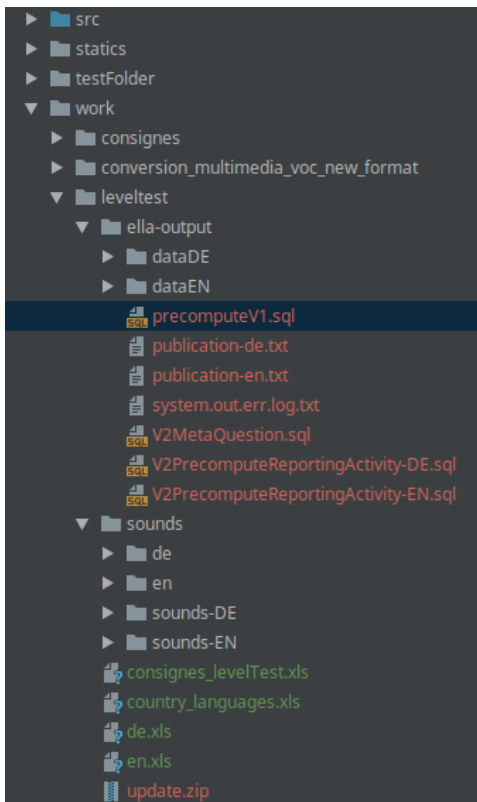
<leveltest.publish.question.excel lg="cs" lgUP="CS"/>
<leveltest.publish.question.excel lg="da" lgUP="DA"/>
<leveltest.publish.question.excel lg="el" lgUP="EL"/>
<leveltest.publish.question.excel lg="pl" lgUP="PL"/>
<leveltest.publish.question.excel lg="pt" lgUP="PT"/>
<leveltest.publish.question.excel lg="sv" lgUP="SV"/-->

```

When it's done, you can launch the publication script.

Check the publication output

You should see this kind of content :



The ella-output folder should contain a folder for each language you want to publish like dataDE or dataEN, some sql files and some log files for each language.

Check if there is any error in the logs files. If there isn't any error, well done!

In case you want to export sounds : **Check if the sound folders are filled with the sounds.** ex: the sounds must be in dataDE/sounds.

/!\ The script concats sounds when needed! It's very common that there is less sound after the script has processed

Rename and put the update zip on the NAS

Simple step, rename the update.zip file into something like this : "20180411-update-leveltest-lang-A.zip" and copy the file in **/data/newSources /leveltest/**

Tell Greg and J-P that the export is done and ready to publish

Finally, just tell them that the zip file is ready for the update and if there are sounds to update. Easy game, GG, your job is done here!