

Automatic daily source string updates for Audacity

The following code and its description are thanks to **Michele Locati**. The code allows for daily updates of the Audacity .pot file (source strings), allowing translators to translate practically in real time, independent of *string freezes* and release schedules.

The 'unofficial' Transifex project for Audacity is used to manage and update translations: <https://www.transifex.com/klyok/audacity/>

Github repository: <https://github.com/mlocati/audacity-pot>.

Overview (credits: Michele Locati)

GitHub offers a great feature: GitHub Actions. That's a tool that you can use to create super fancy stuff, letting users do something when a certain event occurs. The cool thing is that everything is hosted on GitHub, so we don't need any server.

In this particular case:

<code>name: Update .pot</code>	
<code>on:</code>	
<code> schedule:</code> <code> - cron: "0 8 * * *"</code>	the script is executed at 8 am UTC every day because of these lines
<code> repository_dispatch:</code> <code> types:</code> <code> - update-pot</code>	these lines allows me to manually trigger the execution, useful for example when I change something in the script and I want to see if it works without having to wait for the 8am
<code>jobs:</code> <code> update-pot:</code> <code> name: Update audacity.pot</code> <code> runs-on: ubuntu-latest</code> <code> steps:</code>	
<code> - name: Install required packages</code> <code> run: sudo apt-get install -qy gettext</code>	these lines installs the required system tools needed by the Audacity script that regenerates the <code>audacity.pot</code> file

<pre>- name: Checkout audacity-pot repository uses: actions/checkout@v2 with: fetch-depth: 0</pre>	<p>these lines create a local copy of my audacity-pot repository on the server that's executing the script</p>
<pre>- name: Switch to main branch run: git checkout --quiet main</pre>	<p>with these lines we ensure that we use the right branch of the audacity-pot repository</p>
<pre>- name: Cleanup run: rm -rf tmp && mkdir tmp</pre>	<p>with these lines we create a temporary directory</p>
<pre>- name: Checkout audacity repository uses: actions/checkout@v2 with: repository: audacity/audacity ref: master path: tmp/audacity</pre>	<p>with these lines we "download" a copy of the audacity repository</p>
<pre>- name: Regenerate audacity.pot run: set -o errexit cd tmp/audacity/locale AUDACITY_ONLY_POT=y ./update_po_files.sh cd - >/dev/null cp tmp/audacity/locale/audacity.pot .</pre>	<p>with these lines we regenerate the <code>audacity.pot</code> file by calling the update po files.sh script provided by the Audacity repository</p>
<pre>- name: Check and commit changes env: DEPLOY_KEY: "\${{ secrets.DEPLOY_KEY }}" run: </pre>	
<pre> printf 'Checking changes... ' if test -z "\$(git diff --unified=0 --no-indent-heuristic --no-color audacity.pot grep -E '^[+-]' tail -n+3 grep -vE '^[+-]#' grep -vE '^[+-]'POT-Creation-Date)"; then echo 'no changes detected.' exit 0 fi</pre>	<p>with these lines we check the just generated <code>audacity.pot</code> changed vs the <code>audacity.pot</code> file already present in the audacity-pot repository (without considering changes in comments and in the date/time when the <code>.pot</code> file has been generated). If no change is detected the script ends</p>
<pre> echo 'changes detected.'</pre>	

```
printf 'Configuring git... '
git config user.name 'Michele Locati'
git config user.email 'michele@locati.it'
echo 'done.'
echo 'Committing changes:'
git add audacity.pot
git commit -m 'Update audacity.pot'
printf 'Initializing SSH agent... '
eval "$(ssh-agent)" >/dev/null
printf '%s' "$DEPLOY_KEY" | tr -d '\r' | ssh-
add - >/dev/null
echo 'done.'
echo 'Pushing changes:'
git push origin main:main
```

with [these lines](#) we save the newly generated audacity.pot file to the audacity-pot archive (you can see the history of these changes [here](#))