GeekSpeak

Jost Zetzsche
jzetzsche@internationalwriters.com

The GeekSpeak column has two goals: to inform the community about technological advances and at the same time encourage the use and appreciation of technology among translation professionals. Jost also publishes a free technical newsletter for translators (www.internationalwriters.com/toolkit).

Cultural Rifts

I have written before about the strange phenomenon surrounding translators of open-source material, who, though by most accounts often very technical, have no technology intersection with their commercial (and often much less technically inclined) counterparts. Virtually none of the tools used by open-source translators are used by commercial translators, and vice versa. There are exceptions, of course. The open-source translation environment tool (TEnT) OmegaT, built mostly for and by professional commercial translators, is used by some open-source translators, and there are a handful of commercial translators using tools like Virtaal, Pootle, and Poedit. But the crossover is the exception rather than the rule.

Virtually none of the tools used by open-source translators are used by commercial translators, and vice versa.

There are reasons for this discrepancy. First, the commercial TEnTs are, well, commercial, pricey, and therefore not accessible for the mostly volunteer open-source translators. This becomes even more relevant when you consider that most open-source translators work in real-time workgroups[1]—a feature that in most commercial products REALLY costs a lot. In addition, most commercial TEnTs used to run on Windows only, which was not particularly popular among many connoisseurs/translators of open-source software products. And finally, this serves only to highlight the cultural rift between the two groups, one that in my opinion is unfortunate.

There is one other reason as well, though, and that is different file formats. Many open-source projects are not based on the commercial file formats on which yours and mine are based, but on ones that many of us may never have heard of before. The formats that particularly come to mind are the GNU gettext .po and .pot files.

Years ago, as I was writing the documentation for a TEnT, I was stunned when I had to document that particular filter. I had never seen a file like that, nor did I know what the files were even good for! Fortunately, one of the few translators who "lives in both worlds" took me under his wing, sent me sample files, and explained to me what the files were about. Eight or nine years later, I now have to work in projects based on .po files—something that is becoming increasingly popular, especially in newly released software products with developers who originally came from the open-source corner.

So, what are these files? These are the translatable language resource files used in the free GNU gettext concept for translating software and documentation. GNU gettext is the de facto standard in many open-source projects, and it works with a large variety of programming languages.

In many ways, .po files are quite sophisticated. Long before XLIFF existed or the Trados TagEditor .ttx format had become a standard of sorts, these bilingual translation files were the actual translation files and could themselves become translation memories for other projects. All this was made possible by the above-mentioned Poedit, an open-source translation editor. You might have noticed that I do not call it a TEnT—that would be saying too much. It has the definite feel of a Notepad compared with Word or OpenOffice (no fuzzy matching, no spell-check, no concordance searches, no terminology components, etc.).

There are some things that can be done very well with it. One is to turn .pot files (the template files that are directly derived from the software) into the bilingual .po files, and the other is to compile the translated .po file in its compiled .mo version—both extremely welcome features when you receive .pot files and are asked to deliver translated .mo files.

As far as translation goes, it is interesting for experience's sake to take on one small project in Poedit, but you will want to go back to your good old complete translation environment quickly. Which tools support this format? Here is a list of TEnTs that directly support it:

- Déjà Vu;
- Heartsome;
- Swordfish;
- SDL Passolo (a localization tool); and, not surprisingly,
- OmegaT, plus a number of other open-source tools that tend to have the same limitations as Poedit.

This is great if you have one of these tools for your ongoing project, but what if you do not?

Enter Rainbow, the tool that will do what no other can, the tool that will go the extra mile, the tool whose developers, under the leadership of industry veteran Yves Savourel, do all those things that others are too lazy to do or do not have time to do.

Rainbow allows you to carry out a multitude of localization-related tasks (see its wiki page under the links listed in the box), but its most important function in our context may be the preparation of translation packets for XLIFF, OmegaT, and Trados RTF for a great number of otherwise not supported or badly supported file formats (including .po).

To convert your .po files to either XLIFF or Trados RTF where the non-translatable text is protected with the Trados tw4winExternal style, follow these steps:

1. Download, install, and start Rainbow.
2. Select **Input > Add Document** to insert files and select the .po file(s).
3. Select **Utilities > Translation Package Creation** and choose either **Generic XLIFF** or **Original + RTF Layer**.
4. Select a location on the **Package Location** tab.
5. Press **Execute**.

You will now have a folder structure that contains a Work folder in which you can find the files that need to be processed with the TEnT of your choice. (And if your .po file was already partly translated, courtesy of Yves and his team, a translation memory exchange TMX file would have also been created with that content right above the Work folder.)

Once you are done with the translation:

1. Save the files back into the Work folder.
2. Start Rainbow again.
3. Select **Input > Add Document** to insert files and select the manifest.xml file above the Work folder.
4. Select the translated file(s).
5. Select **Utilities > Translation Package Post-Processing**.
6. Select **Execute**.

You can now pick up your translated .po files in the Done folder. And if you need to deliver .mo files, open them in Poedit and simply save them. In the default setting, Poedit will have created an .mo file for you.

(If you are thinking you will never need to know this information, keep this in mind: Hosni Mubarak expected his 2011 to go a little differently than how it panned out, too. So keep these instructions in a safe place!)

Note

1. Thanks to members of the OmegaT user group (http://tech.groups.yahoo.com/group/OmegaT/) for bringing this and other things to my attention.

Links

OmegaT
www.omegat.org

Poedit
www.poedit.net

Virtaal
translate.sourceforge.net/wiki/virtaal

Rainbow
www.opentag.com/okapi/wiki/index.php?title=Rainbow

Pootle
translate.sourceforge.net/wiki/pootle