

# Translating AbiWord

## Getting started

- 1 If you want to work on a particular language, say Pelonian (language **zz-ZZ**), you may start this way:  

```
/update.pl --pot  
mv abiword.pot zz-ZZ.po
```

(The .po suffix doesn't mean Pelonian; it applies to any language.) Now edit **zz-ZZ.po**. Fix the comment at the start to indicate who you are, what the language is, and when you are doing this work.
2. I strongly recommend you use Unicode UTF-8 to encode all your Pelonian strings. In the comment at the top of **zz-ZZ.po**, make sure the following two lines appear:  

```
"Content-Type: text/plain; charset=UTF-8\n"  
"Content-Transfer-Encoding: 8-bit\n"
```
3. Now you want to translate all lines that start with **msgid** and place the translation in place of the empty string in the following msgstr line. For instance, you might translate  

```
msgid "the dictionary"  
msgstr "grompabikku-la"
```

(Sorry, my knowledge of the Pelonian language is limited.) Translate as many entries as you like. Entries that have **%d** or **%s** should most likely be translated with **%d** and **%s** in place, referring to numeric or string data to be filled in later. Don't worry that lines get very long. If you like, you may divide long lines:  

```
msgstr "  
    "implog revulam"  
    "groppsigu moturo nula"
```

Later processing steps will automatically divide long lines for you in any case.
4. Once you have translated as much as you want, check your translation for sanity:  

```
msgfmt -cv zz-ZZ.po
```

Fix any errors that msgfmt finds. Then run this script:  

```
./ui-backport.pl zz-ZZ
```

This program places strings in **../user/wp/strings/zz-ZZ.strings**. The next time you build AbiWord, the Pelonian strings will be available, or you can manually copy the strings file to **\$prefix/AbiSuite-2.0/AbiWord/strings/zz-ZZ.strings** and see them immediately when you next run AbiWord.
5. Later, you will want to fix your translations, finish strings you didn't translate earlier, and see if new strings have been introduced into AbiWord that you need to translate. Run this script:  

```
./update.pl zz-ZZ
```

You will see a summary that tells you how many messages have been translated and how many are still untranslated. It will also indicate how many "fuzzy" translations there are. These are translations that you have made but which may no longer be right, because AbiWord no longer uses quite the same text in the message that it did when you first made the translation. Search for the word "fuzzy" in **zz-**

**zz.po**, and fix all such translations, if necessary. Remove the line that says "fuzzy" to indicate that you have fixed them. Then do whatever other translations you want, and backport as indicated above.

6. There are several alternative techniques for building Pelonian string files, although I don't recommend them.
  1. Look in <http://www.abisource.com/dev/strings/strings.html> (quite long; all you need is the beginning) to find a fairly complete translation of some language that you can work from, perhaps because the language is related to Pelonian. Say you choose **no-NO**. Copy **user/wp/strings/no-NO.strings** to **user/wp/strings/zz-ZZ.strings** and modify the translations. Disadvantages: (a) You can't easily keep up with new strings added to AbiWord over time, (b) You can't easily find which Pelonian strings you haven't done yet, (c) In the cases where the same English string occurs in multiple instances, you need to introduce your Pelonian translation for each instance. (d) You have to trust the no-NO translation, which is one step removed from the English source.
  2. Run **AbiWord --dumpstrings**; you will find a fresh file in the current directory called **en-US.strings**. Copy this file to **user/wp/strings/zz-ZZ.strings** and edit it. Disadvantages: (a-c) above.

## Unix and Gtk+ issues

1. In order for your translations to appear, you need to set your environment to specify your language. Before you start AbiWord, set your **LANG** and **LC\_ALL** environment variables to

```
LANG=zz-ZZ.utf8
LC_ALL=zz_ZZ
```

AbiWord uses the **LANG** variable; Gtk+ uses the **LC\_ALL** variable.

When you run AbiWord, you should see all the strings you have translated in Pelonian, not in English.

2. You might not like the font in which Pelonian is displayed. In your home directory, place a file called

```
.gtkrc-2.0
```

In that file, place a line saying

```
gtk-font-name = "sans 20"
```

The 20 is a size indicator; you may change that if you like. Now you must tell Pango (which actually draws the fonts) what you mean by sans 20. Copy

**/usr/local/etc/pango/pangox.alias**es (it may be in a different place, such as **/etc/pango**) to **~/.pangox.alias**es). Add your favorite Pelonian font (mine is **-misc-fixed-medium-r-normal--20-200-75-75-c-100-iso10646-1**) to the top of the entry for "sans normal normal normal normal". You may have to remove other fonts from that list if you don't get the font you want. If your favorite font is monospace (like mine), set your **gtk-font-name** to **"monospace 20"** and put your font name in the entry for **"monospace normal normal normal normal"**.

If you don't see any font at all, your Unix machine is most likely using the fontconfig library. Look at its configuration file in **/etc/fonts/fonts.conf** to see where

it finds fonts. For instance, you might find that it mentions the directory **/usr/share/fonts**, which is a good place to put extra fonts, such as Unicode TrueType fonts. Apparently **/usr/share/fonts** must not be a symlink to a remotely mounted filesystem.

3. Your AbiWord translations only apply to those menus and buttons that AbiWord builds for itself. AbiWord also uses "stock" items from Gtk+, including buttons that say "Cancel" and "OK". If you don't see these messages in Pelonian, you need to introduce Pelonian strings to the gtk+ library. Get the source distribution of gtk+ (mine is **gtk+-2.0.0** as of 2/2003). In the **po** subdirectory, run

```
./update.pl --pot  
mv abiword.pot zz_ZZ.po
```

and edit the **zz\_ZZ.po** file just as you did for Abiword, paying special attention to those items labeled with **gtk/gtkstock.c**.

Modify the Makefile, adding **zz\_ZZ.po** and **zz\_ZZ.gmo** everywhere you see **zh\_CN.po**.

```
make zz_ZZ.gmo
```

```
make install (you may need privilege)
```

4. If you get the error message that your **zz\_ZZ** locale isn't recognized, or you don't see stock items in Pelonian, try

```
locale -a
```

If **zz\_ZZ** is not listed, you need to set up Pelonian as one of the supported locales on your machine. You will need privilege, typically as root.

```
cd /usr/share/i18n/locales/
```

[pick any language that seems similar to Pelonian, such as **en\_GB**]

```
cp en_GB zz_ZZ
```

[edit **zz\_ZZ** as needed, or just leave it the way it is]

```
localedef -i zz_ZZ -c -f UTF-8 zz_ZZ
```

[for older versions of localedef]

```
localedef -c -f UTF-8 \  
-i zz_ZZ /usr/share/locale/zz_ZZ
```

```
localedef --add-to-archive /usr/share/locale/zz_ZZ
```

[for newer versions]

5. If Pelonian is written right-to-left (RTL) instead of LTR, you might need to put a directional marker such as **u202b** at the start of any message that includes non-RTL characters, such as periods, query marks, and numbers. But a better solution is to set the translation for the Gtk (not AbiWord) **msgid "default:LTR"** to be **msgstr "default:RTL"**. This single setting modifies the "outer" directionality, influencing the position of widgets within containers and the default direction of all text.