FieldWorks Localization

Ken Zook

October 28, 2008

Contents

1 Introduction

Provision has been made for localizing most strings in the user interface for FieldWorks Language Explorer and Translation Editor. Data Notebook and List Editor do not yet have this capability. It is also possible to add localizations to the semantic domain list.

Here are some limitations with the current strategy.

- The size of dialog boxes do not change, so longer strings may cause problems.
- Dialogs written in C++ (e.g., Backup/Restore, File Open) cannot be localized.
- You can only localize to languages supported by the operating system.
- Each string has a single translation, so it can’t be translated differently in different contexts.
- We can’t localize icons and other graphics.

2 Localizing the User Interface

2.1 Overview

The localization process involves coordination between the development team and localization translators, typically located in other countries. The major disadvantage with this approach is the translator cannot see the results of their work until they receive a new installer or set of files.

This summarizes the localization process.

1. A developer produces a POT translation file based on the current FieldWorks source files. This file is sent to the translator.
2. The poEdit program is used to convert this into a PO file, or update an existing PO file for a specific language.
3. The translator uses poEdit to add translations to this PO file.
4. The PO file is sent to a developer.
5. The PO file is merged with any existing PO file for that language.
6. The nightly build process uses the PO file to generate a localized set of DLLs and a localized strings XML file.
7. The installer includes these localized files
8. Anyone installing FieldWorks via the installer will then be able to use the localized version.

Flex and TE currently have independent UI languages.

To change the UI language in Flex, use Tools…Options, in the Interface tab choose the User interface language and the UI changes immediately.

To change the UI language in TE, use Tools…Options, pick the General tab and choose the User interface language, then restart the program.

Doug Higby has set up a FieldWorks translation project that is accessible from the web at https://translations.launchpad.net/fwlex and https://translations.launchpad.net/fwte. Multiple translations can be made at this site.

2.2 What does a developer do?

2.2.1 Creating a POT Translation file

At any time a developer can generate a POT file to send to a localization translator. This process extracts all translatable strings from resx files and xml files under Language Explorer\Configuration. The command to generate a POT file from the fw\bin directory is

```
LocaleStrings –x
```

This assumes you want to extract data for all programs and that your RootCodeDir string variable in HKLM\Software\SIL\FieldWorks is set to the current fw directory. (If RootCodeDir is not set, it will use the fwroot environment variable if it is set.) If not, then you'll need to use other command line options (e.g., -r, --no-flex, --no-te, --pot). The default output file will be FieldWorks.pot in the fw\bin directory.

2.2.2 Integrating translations into the build process

When a translator returns a PO file for a given locale, it needs to be added into our Perforce repository. If this is the first Messages.xx.po file for a given locale (xx represents the locale identifier—see list at http://msdn2.microsoft.com/en-us/library/system.globalization.cultureinfo.aspx), it can simply be checked into the fw\Localizations directory. If there is already a Messages.xx.po file for this locale, you'll need to merge the changes from the new PO file to the existing file. To do this, use the following command.

```
LocaleStrings -m oldfile.xx.po newfile.xx.po
```

This process will back up the original file (e.g., messages.fr.po-20070524120146) and then merge the information into the original file. It also generates a log file (e.g., Merge-20070524120146.log) listing any conflicts. The log file contains a summary of the changes at the end of the file.

To manually build the localized files, use the following Nant target

```
nant [release] localize
```

If you want to build release versions, be sure to include the ‘release’ nant target before the ‘localize’ target. This process creates individual output\xx directories containing translatable resource files. These are then compiled into localized dlls in output\debug\xx. It also creates a strings-xx.xml file in DistFiles\Language Explorer\Configuration for localizing the Flex XML strings.

The overnight build process creates localized versions and currently adds the fr localizations to the installer. At a future time the installer will provide options to let the user decide which localizations to install. On an end-user machine the output\release\xx directory is stored as a FieldWorks\xx directory, and strings-xx.sml is stored in FieldWorks\Language Explorer\Configuration.

2.2.3 Upgrading a user .po file to a new version

When a translator returns a PO file for an older version and you need to build dlls for a new version and want to incorporate the new strings in the po file, use the following steps.

1. Use 2.2.1 to create a new pot file for the new version.
2. Open the user’s po file with poEdit and use Catalog…Update from POT file to add in the newly created POT file.
3. Edit the updated .po file changing the Project-Id-Version to the new FieldWorks version.
4. Follow 2.2.2 to generate dlls to return to the user.
5. You can either return the newly created po file to the user (if he hasn’t done any work in the meantime) or send the pot file and let him merge the new file with his.

2.3 What does a translator do?

PoEdit is a free translation program you can download from the Internet and use to translate source language strings into target language strings. It provides many useful features for doing this translation including translation memory.

2.3.1 Initial setup

To prepare for translation, you need to install the poEdit program and create a catalog for the target language. These are the steps to do this.

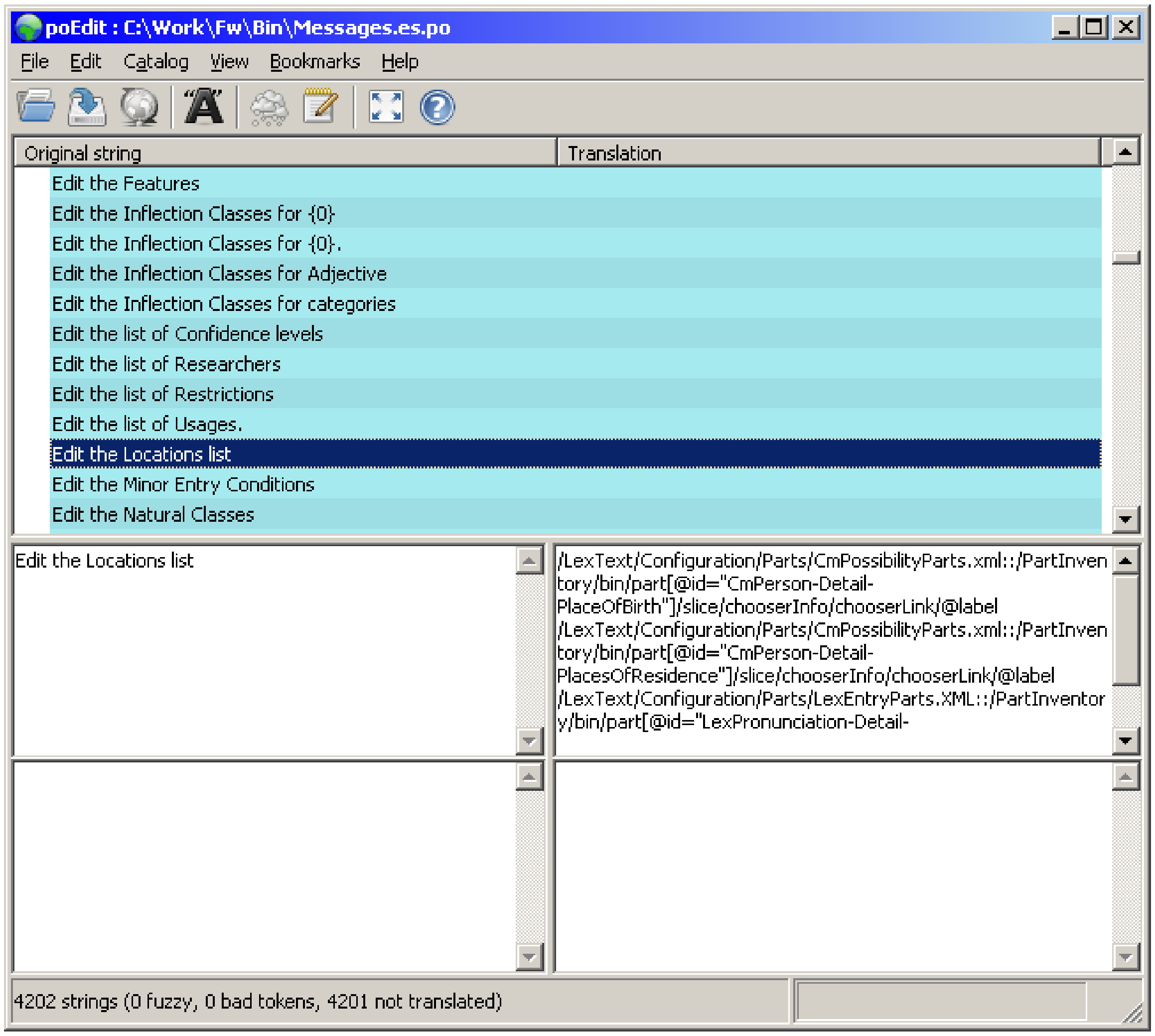
1. Download and install poEdit from http://www.poedit.net/download.php.
2. Run poEdit the first time and answer initialization information.
3. Use File…New catalogue from POT file… and select the POT file you received from the developers. When it asks for information, fill in answers similar to the following
   Project name and version: FieldWorks N.N
   Team: your name(s)

Team's email address: your e-mail address
Language: the target language for translation
Country: the target country
Charset: utf-8
Source code charset: utf-8

This process creates a Messages.xx.po file where xx is the target system language designator you have chosen, such as 'es' for Spanish.

2.3.2 Using poEdit

During the translation process, you start poEdit and use File…Open to open the PO file you created.


poEdit : C:\Work\Fw\Bin\Messages.es.po
File Edit Catalog View Bookmarks Help
Original string
Translation

Edit the Features
Edit the Inflection Classes for {0}
Edit the Inflection Classes for {0}.
Edit the Inflection Classes for Adjective
Edit the Inflection Classes for categories
Edit the list of Confidence levels
Edit the list of Researchers
Edit the list of Restrictions
Edit the list of Usages.
Edit the Locations list
Edit the Minor Entry Conditions
Edit the Natural Classes

Edit the Locations list

/LexText/Configuration/Parts/CmPossibilityParts.xml::/PartInventory/bin/part[@id="CmPerson-Detail-PlaceOfBirth"]/slice/chooserInfo/chooserLink/@label
/LexText/Configuration/Parts/CmPossibilityParts.xml::/PartInventory/bin/part[@id="CmPerson-Detail-PlacesOfResidence"]/slice/chooserInfo/chooserLink/@label
/LexText/Configuration/Parts/LexEntryParts.XML::/PartInventory/bin/part[@id="LexPronunciation-Detail-

4202 strings (0 fuzzy, 0 bad tokens, 4201 not translated)

The top pane shows the strings in the catalog. Information for the selected string is shown in the bottom panes. The left contains the source (top) and target (bottom) strings the right contains source (top) and target (bottom) notes. The source notes come from the POT file and give information on the source of the string. You can put any information you want in the target note field. To do this, click the Edit comment button in the toolbar.

You can keep track of uncertain translations by using the Fuzzy Translation toggle button on the toolbar.

Some strings contain templates with placeholders for substrings. For example ‘Edit the Inflection Classes for {0}.’ During operation, the program will insert a substring into the string at at the {0} location (e.g., ‘marker’). You can move the placeholders around to any location in the string template. Where more than one placeholder is present, you can reorder these as needed. The substring for {1} will be inserted wherever {1} occurs in the template.

Some strings are designed for menus. They have an underscore preceding the character that is used as an Alt key shortcut. For example the string ‘_Complex Feature’ will be displayed in a menu as ‘Complex Feature’ and can be activated by Alt+c. You can change the location of the underscore to change the shortcut key, but you should be careful that a given letter is not underlined more than once in any menu.

All of the changes you make are stored in the PO file. When you want to see the results of your work, you’ll need to send the PO file to a developer to incorporate into the next nightly build.

Note: At least for FieldWorks 5.4 and earlier, there is an ‘en’ string in the localization file. This string should be translated to the locale string of the language into which you are translating. For example, if you are translating into German, this should be ‘de’. Any other string can cause problems, particularly with TE File…Open.

2.3.3 Merging an updated POT file

As the program changes, you may receive a new POT file from developers where new strings have been added, others have been changed, and some deleted. To continue translations, you’ll need to merge the POT file into your PO file. This is done using the Catalog…Update from POT file… This will add any new strings from the POT file and delete any strings that are no longer in the POT file.

Caution: If you received a FieldWorks.pot file initially and the update is a Flex.pot file, you would lose all of the translations for TE. Before merging files, make sure your original PO file is backed up in a safe place where you can restore it if needed.

2.3.4 Translation memory

PoEdit provides a way to store translated strings in a place where common translations can be applied to any PO file. This is called translation memory. You can set up Translation Memory as follows:

1. Choose File…Preferences and go to the Translation Memory tab
2. Click Add to add the abbreviation for your target translation
3. Click ‘Generate database’ to update the translation memory.
4. In the ‘Update translation memory’ dialog, select the path(s) to search for translation files. The poEdit default is c:\Program Files\poEdit\share\locale.
5. Click Next to select files to use as the source of the translation memory.
6. Click Finish to finish building the translation memory.

You can use the Translation Memory to automatically add translations to your PO file by choosing Catalog…Automatically translate using TM. Translations added in this way are displayed in yellow with a computer icon to the left. You can click the Fuzzy toggle to change these to normal translations.

Microsoft provides 12,400 common English words in some 50 languages in a downloadable file at: http://www.microsoft.com/globaldev/tools/MILSGlossary.mspx. This could be a valuable starting point.

2.3.5 Creating a local strings-xx.xml file

For localizing Flex, it's possible to generate strings-xx.xml directly on your machine without sending your po file to Dallas for processing. This covers most of the menus, tools, and labels in the detail view. It does not cover dialogs. To see dialog changes you'll need to send your po file to Dallas for processing on a full development system.

If you want to generate strings-xx.xml, request LocaleStrings.exe from Dallas. When you get it, copy it into your c:\Program Files\SIL\FieldWorks directory along with a copy of your messages.xx.po file, then give the following command in a command window:

```
LocaleStrings -s c:\Program Files\SIL\FieldWorks\messages.xx.po
```

You should then see a new strings-xx.xml file in c:\Program Files\SIL\FieldWorks\Language Explorer\Configuration, and after restarting Flex, you should see the changes in the UI.

3 Localizing the Semantic Domain list

3.1 Overview

At this point we have a separate approach for localizing the semantic domain list. This process involves adding translations to a master standard format file and then using some processes including CC to convert this into a .SQL file. The SQL file can then be executed on any machine to add these translations to a specific database.

It's also possible using an extension to this process to prepare a localized version of the semantic domain file (Templates\SemDom.xml) that can be added to an end user machine so that it will automatically be included in any new FieldWorks they create on that machine.

To see translations in the semantic domain list, you need to check the desired writing system as either a vernacular or analysis writing system in the Writing Systems tab of the FieldWorks Project Properties dialog.

3.2 What does a translator do?

A translator needs to obtain the SFM file, SemDomV4Template.db, from Dallas. Here is a small sample from this file illustrating domain 1.1.

```
\is 1.1
\sd Sky
\sdt
\dd Use this domain for words related to the sky.
\ddt
\qu (1) What words are used to refer to the sky?
\qut
\ex sky, firmament, canopy, vault
```

```
\ext
\qu (2) What words refer to the air around the earth?
\qut
```

To add translations, a translator should type the translation in the *t marker following each English string. Note this file should be edited as a UTF-8 file. This can be done in programs such as Word or ZEdit after using Options…ReOpen As…Unicode (UTF-8).

3.3 Converting the SFM file to a SQL script

This process can be done locally, or the SFM file can be sent to Dallas for this processing. To do it locally, request a set of CC tables and instructions to be sent from Dallas, then follow the instructions. The result is a SQL file that can be distributed to end users to install into their databases.

3.4 How do you get translations into a database?

Once you have the SQL script to add translations for semantic domains, anyone can install it by executing the SQL script. To do this

1. Copy the SQL file to your %ALLUSERSPROFILE%\Application Data\SIL\FieldWorks\Data directory. (This example assumes the file name is SDAllLoader.sql.
   Note: %ALLUSERSPROFILE%\Application Data is c:\Documents and Settings\All Users\Application Data on Windows XP and c:\ProgramData on Vista.
2. Open a DOS/Cmd box and type the following command line:
   ```
   db exec "SDAllLoader.sql" "DbFileName"
   ```

If your SQL file is in your FieldWorks\Data directory, you don't need to specify a path. Otherwise you need to specify a complete path to the file in this command. You need to substitute your database file name for DbFileName. This will load the file into your database.