Localisation of pCon.planner Plugins

Table of Contents

1. General

The central format of exchange are “portable object” files (ending: *.po). Those allow you to store translations from a source language to a target language like a one-sided dictionary where generally the English language is used as source language.

Such *.po-files will be converted into “machine object” files (ending: *.mo), which is a binary representation of the dictionary.

The pCon.planner uses *.mo-files to provide localisation to Plugin-UI elements too. Convenient for developer and translator would be using the same files for localisation within the Plugin code.

The `.mo` files needs to be located / named according the following scheme:

```
<PluginDir>\<Locale>.mo

<PluginDir>\<Language>.mo

<PluginDir>\<Locale>\<Plugin>.mo

<PluginDir>\<Language>\<Plugin>.mo
```

Examples

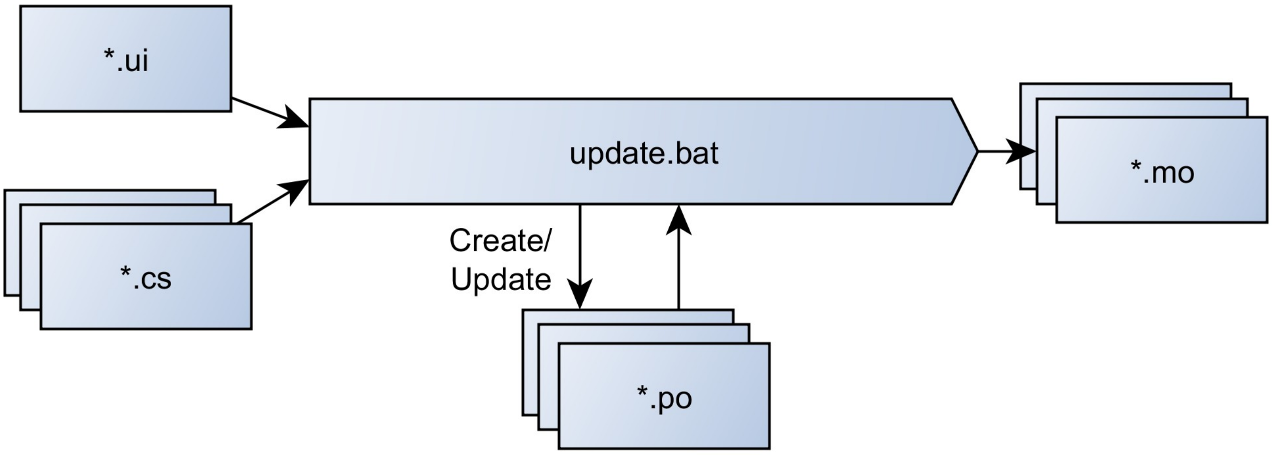
```
\SamplePlugin\en_US.mo

\SamplePlugin\en.mo

\SamplePlugin\pt_PT\SamplePlugin.mo

\SamplePlugin\pt\SamplePlugin.mo
```

2. Recommended Software and Libraries

- Poedit (free), tool to manage translations within “portable object” files
- GNU Gettext (open source), tools to convert *.po to *.mo files
- NGettext (open source), C# library capable of reading *.mo files
- Saxon (open source), tool to extract words from Xml like the *.ui file

3. Scripts for Automation

In your *"SamplePlugin\translation"* folder you'll find a script called *update.bat* which extracts words from the *.ui and all source files of the *SamplePlugin* project. These words are stored as *.po files in "*SamplePlugin\translation\locales*". Existing files will be updated with new vocabularies and can be used for the translation team to work on. The *update.bat* also takes the updated *.po files and builds *.mo files automatically. To modify the amount of languages being used, take a look at *set_languages_var.bat*.



In order to make it work with a different project check the script and adjust the variable *PluginName* accordingly.

To properly function, it is required to set up the tools in your "3rd" folder (The full path is "*pCon.planner Plugin SDK\SamplePlugin\translation\3rd*").

Saxon

For Saxon, the Home Edition is sufficient. Just make sure the *update.bat* can reach it with the command: 3rd\saxon\tools\Transform.exe

GetText

You can get the sources on the given page or use pre-build binaries some websites provide. Ensure the script could execute the commands:

3rd\gettext\bin\xgettext

3rd\gettext\bin\msgcat

3rd\gettext\bin\msgmerge

3rd\gettext\bin\msgattrib

3rd\gettext\bin\msgfmt

NGettext

This Library is required for the *SamplePlugin* to build successfully. You can fetch its sources from the provided link and build it or use our version from "*Plugin Binaries\SamplePlugin\NGettext.dll"*. To make it work copy your or our *dll* to *3rd\NGettext\lib\NGettext.dll*.