

# Internationalization

De Wiki

Aller à : [navigation](#), [rechercher](#)  
[Internationalization](#)

## Sommaire

- [1 The different steps to follow](#)
- [2 Prepare code to translate](#)
- [3 Create a "POT" file](#)
- [4 Create a "PO" file](#)
- [5 Concatenate "po" files" \(optional\)](#)
- [6 Create Java resources](#)
- [7 Create the i18n.properties file](#)
- [8 A good solution to store all these files ...](#)

## The different steps to follow

The internationalization is managed by **GENIUS** using **i18n** standard via the [xgettext](#) toolset, that allows creating message bundles for each language that are accessed in runtime to get the translated strings. Thus, the first point is to get this toolset via [xgettext](#).

Then we have to execute the following steps:

1. Prepare the parts of the code to be translated
2. Extract the strings to be translated inside a "**POT**" file (using [xgettext](#))
3. Create a "**PO**" file (using [msginit](#) or [msgmerge](#)) inside which we will set the translation
4. Eventually concatenate with other "po" files -for example coming from **GENIUS**- (using [msgcat](#))
5. Create a message bundle java file (using [msgfmt](#))
6. At last, create a i18n.properties file to indicate the domain of the message resource bundles

## Prepare code to translate

The translation of the messages in the code is done using the following static methods from [GEnvironment](#). If no translation is available the original message is returned.

Methods Usage:

[GEnvironment.tr\(String message\)](#)

Translate the message message

[GEnvironment.trf\(String formatString, Object... args\)](#)

Translate the format string and then calls to String.format with args

[GEnvironment.trn\(String singularString, String pluralString, long n\)](#)

Translates the singularString and pluralString and returns the translated pluralString if n > 1

Only strings used with these three methods will be extracted by the xgettext tool and stored in the **POT** file ; other strings (identifiers, configuration file names, etc) will not be modified.

At last, we will have to call to the other static method [GEnvironment.initI18n\(\)](#) to specify which

language to use:

```
GEnvironment.initI18n(new Locale("es")); // Spanish language
```

If no parameter is provided (or null argument), no translation is done and we will display what is hard coded.

```
GEnvironment.initI18n();
```

If the locale is wrong or if there is no available po files, the operating system locale will be used. Thus, for example, if the OS locale is "fr":

1. if the french translation file exists, it will be displayed in french
2. if the french translation file does not exist, the display will correspond to what is hardcoded.

```
GEnvironment.initI18n(new Locale("xx"));
```

## Create a "POT" file

To create such **POT** file (here called **test.pot**), we will have to use [[xgettext](#)] as is:

```
xgettext.exe -ktrc:1c,2 -ktrnc:1c,2,3 -ktr -kmarktr -ktrn:1,2  
--from-code=UTF-8 --force-po  
-o po/test.pot $path_java_sources/*.java
```

where \$path\_java\_sources represent the path to go to search for java sources.

## Create a "PO" file

Then, we will create a **PO** file using using [[msginit](#)] if the file does not exist:

```
msginit.exe --input=potFileName --output=poFileName
```

... or [[msgmerge](#)] if it already exists:

```
msgmerge.exe -U poFileName potFileName
```

The suffix of the potFileName will be **.pot** as the suffix of the poFileName file will be **.ll.po** where ll is the *locale* used (fr, en, sp, ...).

Once this **PO** file is created, we will have to modify it to add the right translation. Below is an example of a part of such file (the third string is not translated).

*Note : be careful of the fact that inside the header the predefined character set is not allways put to UTF-8 but sometimes to CP1252 ; in case of it, change it !*

```
"Content-Type: text/plain; charset=UTF-8\n"
```

```
...
```

```
msgid "Button"
msgstr "Bouton"
```

```
msgid "Character string entry:"
msgstr "Saisie d'une chaîne de caractères:"
```

```
msgid "Integer entry:"
msgstr ""
```

If you prefer not to enter directly inside the file, you may also use some utilities as [\[1\]](#)

## Concatenate "po" files (optional)

In case you use other po files (coming from other applications), it is possible to concatenate them using [msgcat](#):

```
msgcat -o finalpoFileName poFileName otherPoFileNames
```

## Create Java resources

The last step is to create Java message resource bundles files with messages that will be incorporated to the application jar file. To generate these resource bundles the [msgfmt](#) command will be used:

```
msgfmt --verbose --java2 -d dirRes -r messFileName -l ll poFileName
```

... where:

- "dirRes" is the directory where the resources will be stored
- "messFileName" is the prefix of the name of the final file (where the .class suffix will be added)
- "ll" is the language (as "en", "fr", "es", ...)

So, the final name will be *dirRes/messFileName\_ll.class*

## Create the i18n.properties file

This file has just to contain one line indicating the name of the resource file.

```
basename=messFileName
```

## A good solution to store all these files ...

In order not to be sure no to loose intermediate files (**POT** file, **PO** file, resource file, ...), a solution is to store all of them in a single directory (for example named **po/**). That is exactly the solution used by the [GIntTool.jar](#) tool. So, to tell to Eclipse to search classes at the right place, you will have just to do this:

1. select your Java projet then click on the right button of the mouse to select *Properties*
2. on the displayed frame, select "Java Build Path" then the *Libraries* tab.
3. push on the "Add Class Folder" button and select the right directory

That's all ...

[Return to the introduction](#) ↑ [Go to the next page](#) →

Récupérée de « <http://genius.cnes.fr/index.php?title=Internationalization&oldid=670> »

## Menu de navigation

### Outils personnels

- [3.141.41.187](#)
- [Discussion avec cette adresse IP](#)
- [Créer un compte](#)
- [Se connecter](#)

### Espaces de noms

- [Page](#)
- [Discussion](#)

### Variantes

### Affichages

- [Lire](#)
- [Voir le texte source](#)
- [Historique](#)
- [Exporter en PDF](#)

### Plus

### Rechercher

<input type="text"/>	<input type="button" value="Rechercher"/>	<input type="button" value="Lire"/>
----------------------	---	-------------------------------------

### GENIUS

- [Welcome](#)
- [Quick Start](#)
- [News](#)

## Basic principles

- [GFrame and GPanel](#)
- [Main widgets](#)
- [Links with Swing](#)
- [GLayout](#)
- [Conditional Display](#)
- [GListener interface](#)

## More deeper in the concept

- [Units management](#)
- [GContainer](#)
- [GReadWrite interface and data files management](#)
- [Modified data](#)
- [Process management](#)

## Still more ...

- [Validity controls](#)
- [Menu bar](#)
- [Icons](#)
- [GClear interface](#)

## Still more again ...

- [Tooltips](#)
- [Shortcuts](#)
- [Copy & paste](#)
- [Plots](#)
- [Results File Management](#)
- [GPlotPanel](#)
- [GGroundPlotPanel](#)
- [Internationalization](#)
- [Log file](#)
- [Update data](#)

## Some other widgets

- [GTabbedPane](#)
- [GTable1D](#)
- [GTable2D](#)
- [GComponentList](#)
- [GDialog and GDetachedPanel](#)
- [GContextFileManagement](#)
- [How to build a standard application](#)
- [GPanTest](#)

- [Create your own widget](#)

## **Evolutions**

- [Main differences between V1.11.4 and V1.12.1](#)
- [Main differences between V1.10.1 and V1.11.4](#)
- [Main differences between V1.10 and V1.10.1](#)
- [Main differences between V1.9.1 and V1.10](#)
- [Main differences between V1.9 and V1.9.1](#)
- [Main differences between V1.8 and V1.9](#)
- [Main differences between V1.7 and V1.8](#)
- [Main differences between V1.6.2 and V1.7](#)
- [Main differences between V1.6.1 and V1.6.2](#)
- [Main differences between V1.6 and V1.6.1](#)
- [Main differences between V1.5 and V1.6](#)
- [Main differences between V1.4.1 and V1.5](#)
- [Main differences between V1.3 and V1.4.1](#)

## **Training**

- [Training slides](#)
- [Tutorials package for V1.12.1](#)
- [Tutorials package for V1.11.4](#)
- [Tutorials package for V1.10.1](#)
- [Tutorials package for V1.9.1](#)
- [Training & tutorials package for V1.8](#)
- [Training & tutorials package for V1.7](#)
- [Training & tutorials package for V1.6](#)

## **Links**

- [CNES freeware server](#)

## **Outils**

- [Pages liées](#)
  - [Suivi des pages liées](#)
  - [Pages spéciales](#)
  - [Adresse de cette version](#)
  - [Information sur la page](#)
  - [Citer cette page](#)
- Dernière modification de cette page le 1 avril 2019 à 10:33.
- [Politique de confidentialité](#)
  - [À propos de Wiki](#)
  - [Avertissements](#)

