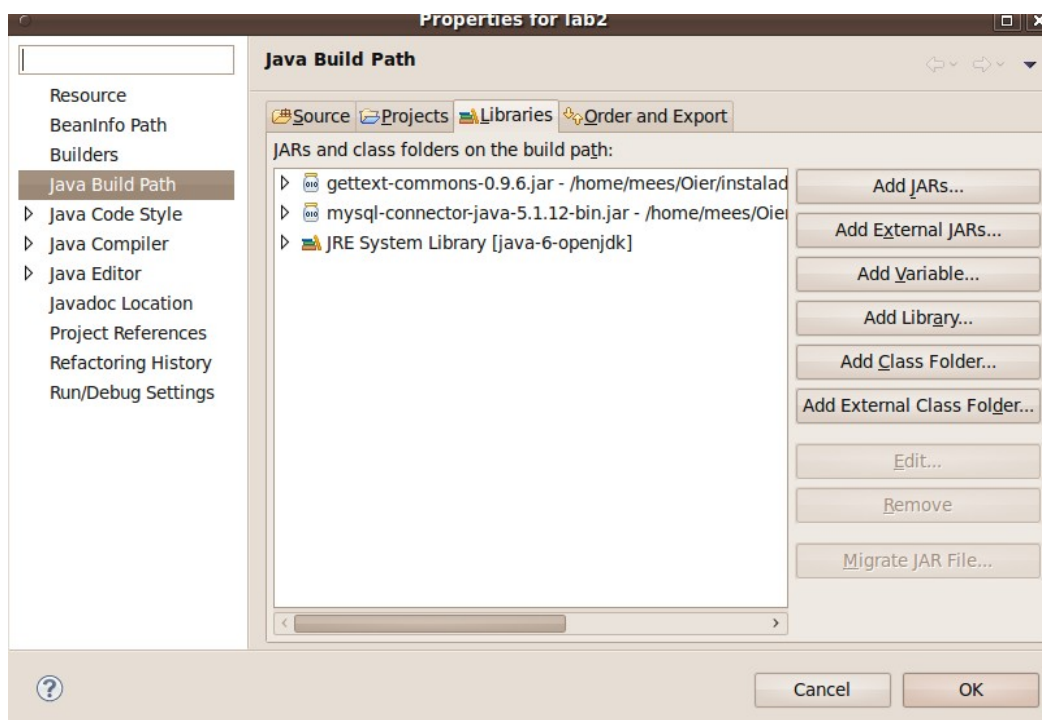


## Java Programak Internazionalizatu

Programak garatzerakoan kontuan hartu behar da, aurkezpen mailan erabilitako hizkuntza ezagutzen ez duten erabiltzaileak egongo direla normalean. Dokumentu honetan, Java programak Eclipse-rekin garatzean hauek nola egin itzulgarriak azalduko da. Honetaz gain, itzulpenak nola egin eta erabili ere argituko da adibide batekin. Javak itzulpenak lantzeko ekartzen dituen tresnak nahiko konplexuak eta erabiltzeko deserosoak direnez Gettext-commons libreria erabiliko da helburu hautarako.

### Lanean hasi aurretik:

1. Gettext-commons jaitsi behar da eta eclipsen zure proiektuan integratu. Helbidea: <http://code.google.com/p/gettext-commons/downloads/list>  
Bertan azken bertsioaren .jar-a jaitsi. **Garrantzitsua** gettext-commons-bertsioZbki.jar jaitsi behar da, ez nahastu ant-tasks edo source paketeekin.
2. Eclipse proiektuan integratu.  
Proiektuan gainean “Properties” aukeratu eta bertan “Java Build Path”. Hor “Libraries” atalean “Add External JAR” botoia sakatu eta aurreko puntuan jaitzitako fitxategia aukeratu.



### Testu kateak itzulgarri gisa markatu:

- Aurkezpen mailan erabiltzaileak ikusiko duen testu kate bakoitza itzulgarria dela adierazi behar da. Horretarako klase bakoitzean I18n motako objektu bat sortuko beharko da. Ez ahaztu “import”-ak egiteaz.

```
import org.xnap.commons.i18n.I18n;
import org.xnap.commons.i18n.I18nFactory;
....
....
I18n i18n = I18nFactory.getI18n(getClass());
```

Orain testu kate bat itzulgarria dela tr funtzioarekin adierazten da:

```
System.out.println(i18n.tr("Textu hau itzulgarria da!!"));
```

edo jlabel bat bada:

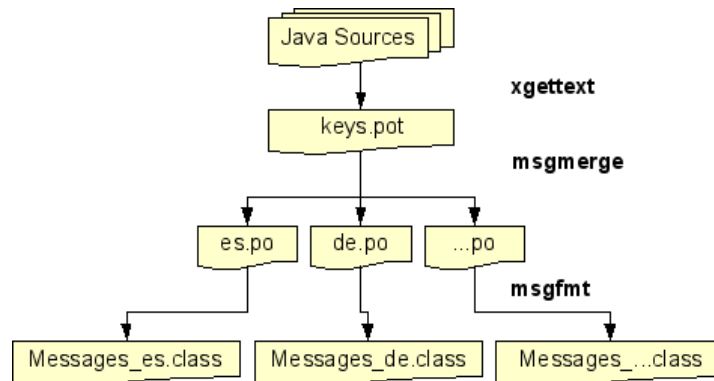
```
jLabel1.setText(i18n.tr("Erabiltzaile izena:"));
```

## Fitxategi itzulgarriak sortu

Programan testu itzulgarri guztiak tr funtzioarekin markatu eta gero Gettext tresnak erabiltzeko garaia da. Linux banaketetan defektuz instalaturik egoten dira normalean, windows-en aldiz Cygwin-ekin eta Gettext-en windows bertsioarekin erabili daitekela ematen du. Hala ere, idazten duenak aukera hori probatu ez duenez, ezin du ziurtatu.

- Xgettext komandoak iturburu kodean tr deiak aztertzen ditu eta hauekin POT fitxategi bat sortuko du.
- Msgmerge-ek POT fitxategi oinarritzat hartuta adierazitako hizkuntza bakoitzerako PO fitxategi bat osatuko du.
- Msgfmt-ek PO fitxategiak erabiliz “java class” fitxategiak sortzen ditu, zeintzuk “ResourceBundle” klasearen hedadura izango diren.

Hurrengo irudi honetan ikusi daiteke prozesua.



Hurrengo komandoak keys.pot fitxategia sortuko du.

```
xgettext -ktrc:1c,2 -ktrnc:1c,2,3 -ktr -kmarktr -ktrn:1,2 -o  
po/keys.pot src/*.java
```

Hau da adibide keys.pot fitxategi baten edukia:

```
# SOME DESCRIPTIVE TITLE.  
# Copyright (C) YEAR THE PACKAGE'S COPYRIGHT HOLDER  
# This file is distributed under the same  
license as the PACKAGE package.  
# FIRST AUTHOR <EMAIL@ADDRESS>, YEAR.  
#  
#, fuzzy  
msgid ""  
msgstr ""  
"Project-Id-Version: PACKAGE VERSION\n"  
"Report-Msgid-Bugs-To: \n"  
"POT-Creation-Date: 2010-03-08 17:46+0100\n"  
"PO-Revision-Date: YEAR-MO-DA HO:MI+ZONE\n"  
"Last-Translator: FULL NAME <EMAIL@ADDRESS>\n"  
"Language-Team: LANGUAGE <LL@li.org>\n"
```

```
"MIME-Version: 1.0\n"
"Content-Type: text/plain; charset=CHARSET\n"
"Content-Transfer-Encoding: 8bit\n"

#: lab2/src/lab2/Aurkezpen.java:79
msgid "Erabiltzaile izena:"
msgstr ""

#: lab2/src/lab2/Aurkezpen.java:82
msgid "Pasahitza sartu: "
msgstr ""
```

Hurrengo aginduak .po fitxategi huts bat sortuko du. Fitxategi honetan egingo dira itzulpenak hurrengo pausoa.

```
touch en.po
```

## Fitxategi itzuli

Hurrengo aginduak en.po fitxategian itzuli behar diren lerroak gehitzen ditu. Bestalde, hasierako pot fitxategian agertzen ziren komentarioak kentzen dizkio.

```
msgmerge -U en.po giltzak.pot
```

Aurreko adibidearen kasuan, hau da en.po-ren edukia:

```
#: lab2/src/lab2/Aurkezpen.java:79
msgid "Erabiltzaile izena:"
msgstr ""

#: lab2/src/lab2/Aurkezpen.java:82
msgid "Pasahitza sartu: "
msgstr ""
```

Orain en.po fitxategia editatu eta itzulpenak egin behar dira eskuz. Horretarako *Emacs*, *Poedit* edo *Gedit* erabili daitezke adibidez. *Emacs* eta *gedit* testu editore orokorrak dira, *Poedit* aldiz po fitxategiak editatzeko bereziki prestatu dago. Hurrengo estekan deskargatu daiteke <http://www.poedit.net/download.php>

Aurreko adibidearen kasuan, hurrengo itzulpenak gehitu zaizkio.

```
#: lab2/src/lab2/Aurkezpen.java:79
msgid "Erabiltzaile izena:"
msgstr "User name:"

#: lab2/src/lab2/Aurkezpen.java:82
msgid "Pasahitza sartu:"
msgstr "Enter password:"
```

Hurrengo aginduak itzulitako fitxategitikan javak ulertu dezaken .class bat sortuko du

```
msgfmt --java2 -d src/conf -r app.i18n.Messages -l en en.po
```

“*ResourceBundle*” klasearen hedadura den *.class* bat sortzen du.

Agindu hau egikaritzatzeko *javac* konpilatzailea instalaturik egon behar da.

-d eta gero zehaztutako direktorioa edozein izan daiteke, baina gero sortutako paketea eclipse proiektuaren bin karpeta sartu behar da. Adibide honen kasuan lab2 “*package*” baten barruan daudenez klase guztiak bin karpeta bi azpikarpeta agertu beharko lirateke, app eta lab2.

## Itzulitakoa erabili

Orain hurrengo aginduarekin deitu daiteke sortu berri dagoen itzulpenari.

```
I18n i18n = I18nFactory.getI18n(AurkezpenKlasea.class,  
"app.i18n.Messages", java.util.Locale.ENGLISH);
```

Kontuan eduki *java.util.Locale.getDefault()* metodoarekin sistemaren hizkuntza zein den jakin daitekela. Beraz 50 hizkuntzetarako itzulpenak edukita, metodo horrekin automatikoki sistemaren hizkuntzara aldatuko lirateke interfaze grafikoaren testu kateak.

Iturriak:

<http://code.google.com/p/gettext-commons/wiki/Tutorial>