



Translate texts for the VIPS web site

By Tor-Einar Skog, Bioforsk Plant health division

Version: 2015-03-03

Table of Contents

Translation system – a brief introduction.....	1
Several sub-applications – several translation files.....	1
For the developer.....	3
Creating a new locale.....	3
Updating translation files.....	3
Compiling translation files.....	3

The VIPS web site is a web application tailored to display forecasts and messages from the VIPS system. Its content is translatable. This document describes how to do the translation.

Translation system – a brief introduction

The VIPS web site is based on the web publishing platform called [Django](#), which uses the widely known translation tool called [Gettext](#). The translation happens inside the application code, where a phrase is defined to be translated, for instance:

```
{% trans "Welcome" %}
```

The application knows which language it should translate this phrase to. If it is Norwegian, it will

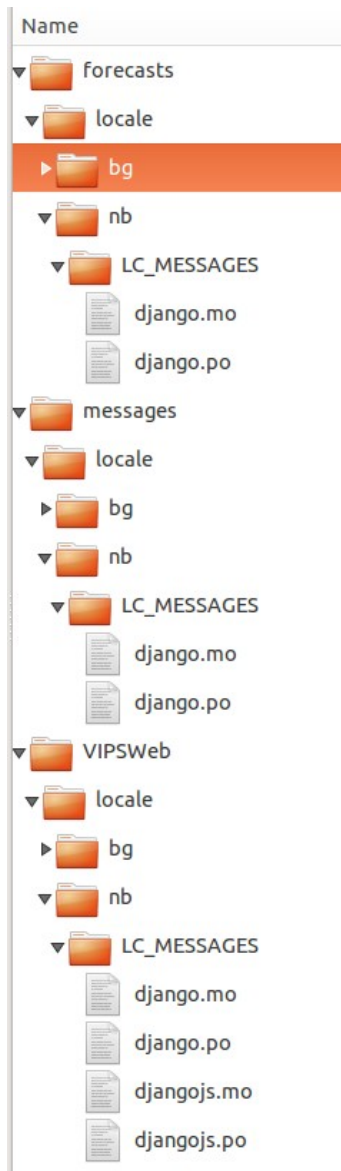
1. look up the phrase «Welcome» in the norwegian translation file
2. find the translation for it in that file. This lookup looks like this:
`msgid "Welcome"`
`msgstr "Velkommen"`
3. return the translation «Velkommen»

If there is no translation file for Norwegian, or the file exists, but the translation for that phrase is not found, the phrase is returned untranslated.

Several sub-applications – several translation files

The web page application contains several sub applications. For the moment these applications are present:

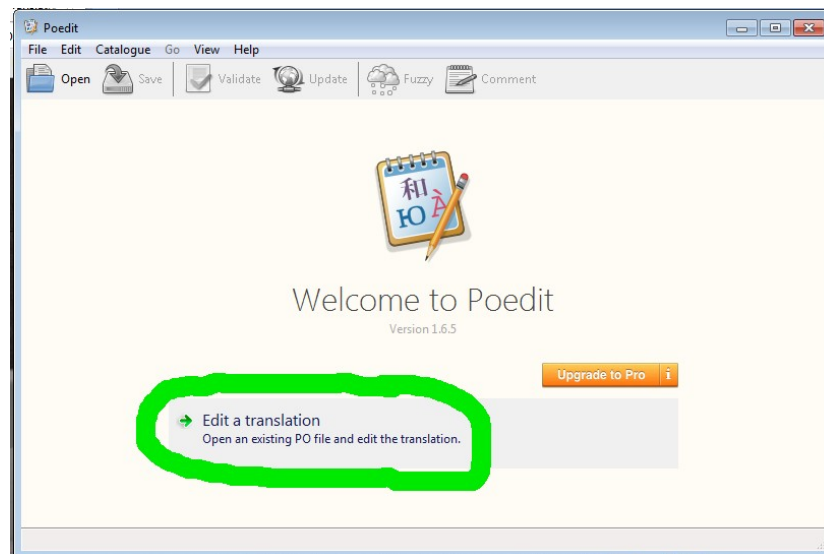
- VIPSWeb – the front page, menu system etc.
- Forecasts – the pages with forecast lists and results



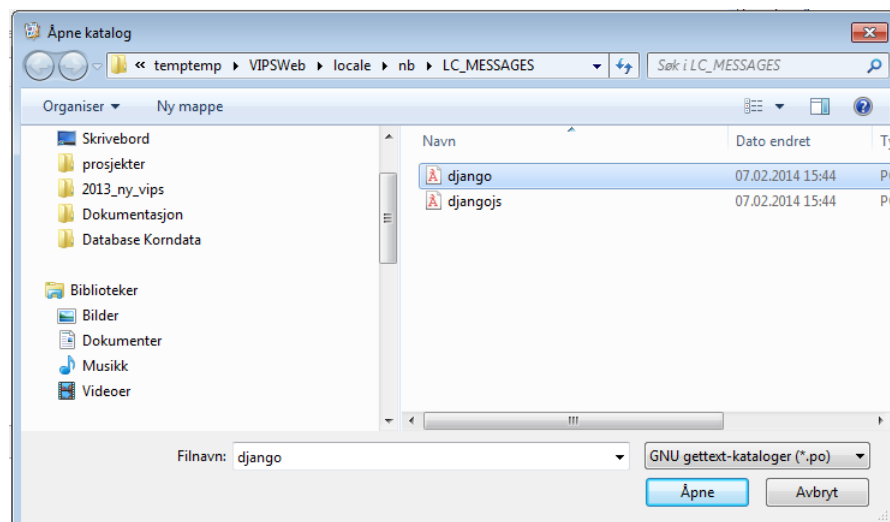
- Messages – the pages with message lists and detailed messages
- Information – a site for publication of a tree structure of documents/documentation

For each application, there is one translation file for each language. So after unZIPing the files and folders, the folder structure thus like this:

So for each application, you have a folder called «locale», which contains folders with each language abbreviated (nb = Norwegian, bg = Bulgarian). Each language specific folder contains a folder called LC_MESSAGES, and inside there you find at least one file ending with .po. And these files are where the translation is done. You can choose to do this with a text editor (like Notepad, jEdit, Emacs, Vim, etc.) - but **NOT MS Word** or any other word processor. OR you could do it the easy way, by using the free program called Poedit. You can download it from here: <http://poedit.net/> . When you start the application, you're welcomed with this:

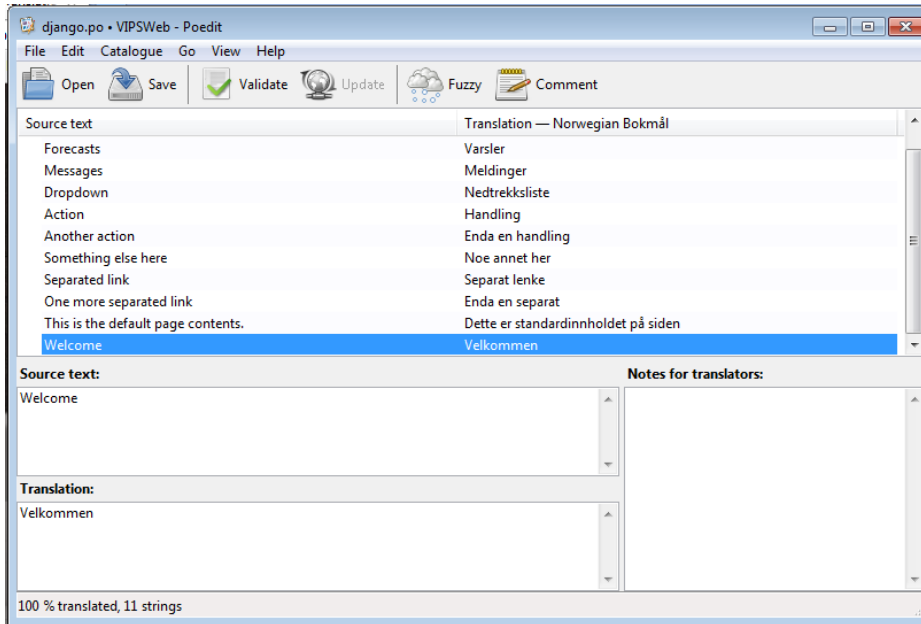


So you choose «Edit a translation», and select your translation file:



After you've opened it,

select each item/phrase and fill in the translation.



After saving the file, the translation is completed. Return the translation by ZIP'ing the whole folder structure and return it by email to Bioforsk (or, if you're an advanced user, you can submit the updated translation files to the source code control system).

For the developer

Creating a new locale

Entering each of the applications (/VIPSWeb, /forecasts, /messages), do:

```
django-admin.py makemessages -l [LANGUAGE]
django-admin.py makemessages -d djangojs -l [LANGUAGE]
```

After that, you can edit the language files (see above)

PLEASE NOTE: django-admin.py has been creating buggy information for plurals. Please check the header field of your newly created .po file. If the plural-forms header does not start with «nplurals=X;», then something has gone wrong. You can find the correct header here: <http://localization-guide.readthedocs.org/en/latest/110n/pluralforms.html>

Updating translation files

If you have added new translation keywords, do this in each of the applications:

```
django-admin.py makemessages -a [LANGUAGE]
django-admin.py makemessages -d djangojs -a [LANGUAGE]
```

Then recompile the translation files (see below)

Compiling translation files

Entering each of the applications (/VIPSWeb, /forecasts, /messages), do:

```
django-admin.py compilemessages
```