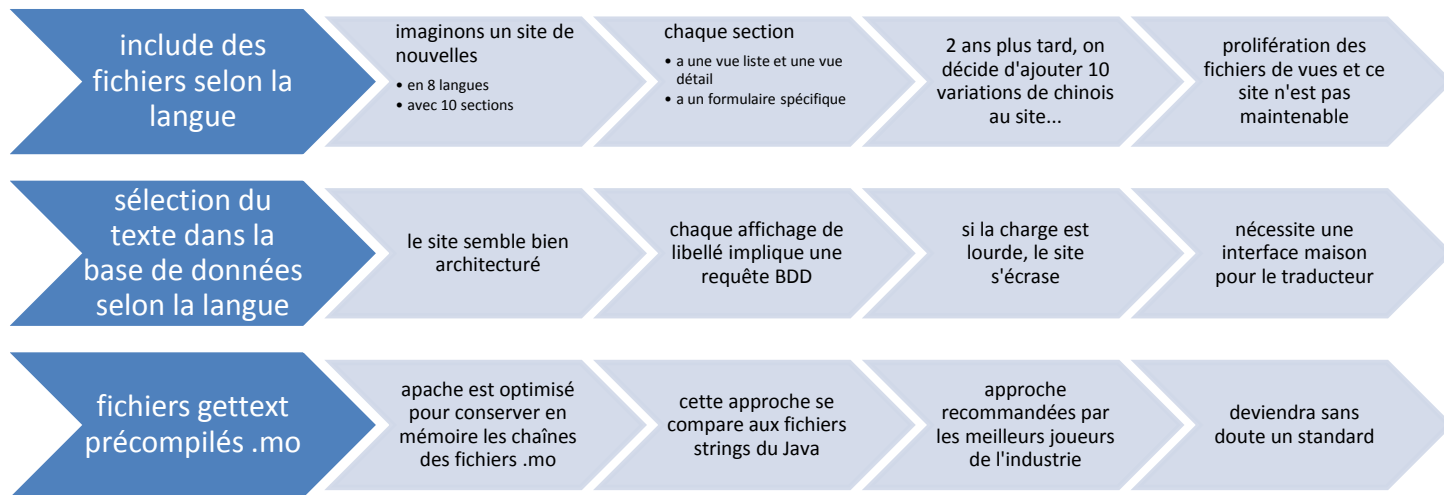


Approches d'internationalisation

Résumé des approches expérimentées



Avantages et désavantages des approches d'internationalisation

	include des fichiers selon la langue	sélection du texte dans la base de données	fichiers gettext précompilés .mo
Architecture	Prolifération des vues Multiples définitions d'un seul visuel... difficile à maintenir.	Parfaite	Parfaite
Performance	La performance n'est pas affectée	La performance est gravement atteinte selon la charge du serveur sauf dans le cloud	La performance est garantie par la prise en charge du standard gettext par Apache
Maintenabilité	Difficile de maintenir les vues pour le graphiste	Un seul fichier par vue Libellés portent souvent une certaine complexité	Un seul fichier par vue. Libellés avec des fonctions minimalistes <?=_('bla')?>
Processus de traduction		Nécessite une interface maison pour permettre la traduction	La maintenance des traductions nécessite un redéploiement des .mo
Robustesse	Assez robuste, chaque vue est testée séparément	Peut afficher une erreur ou encore une chaîne vide, selon la librairie utilisée	Affiche la chaîne originale en l'absence de traduction ce qui est un moindre mal
Célébrités	Personne excepté quelques petits sites rapides et temporaires, c'est une mauvaise approche selon moi	http://www.mediawiki.org/wiki/Localisation https://drupal.org/developing/api/localization http://multilingual-joomla-demo.cloudaccess.net/	http://codex.wordpress.org/Translating_WordPress https://drupal.org/project/gettextapi Des sites professionnels
Firmes web			

En savoir plus sur GETTEXT

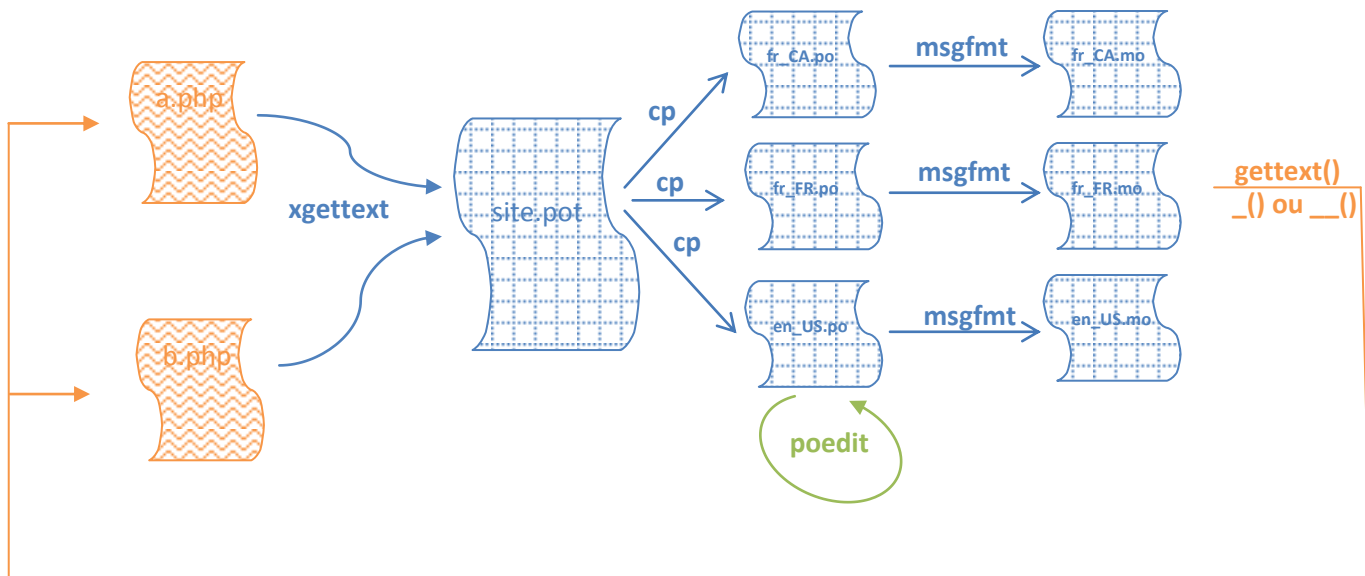
C'est le système de traduction et d'internationalisation standard proposé par GNU

<http://en.wikipedia.org/wiki/Gettext>

<https://www.gnu.org/software/gettext/manual/gettext.html>

Internationalisation d'un site web en PHP

Processus général avec GETTEXT



Lignes de commandes

Lignes de commandes pour un seul fichier php

- 1) `xgettext --from-code=utf-8 /var/www/index.php -o /var/www/locale/monsite.pot`
- 2) `cp /var/www/locale/monsite.pot /var/www/locale/fr_CA.po`
`cp /var/www/locale/monsite.pot /var/www/locale/fr_CA/LC_MESSAGES/monsite.po`
- 3) #Traduire avec poedit ou encore notepad
- 4) `msgfmt -cv -o ca_FR.mo ca_FR.po`

Lignes de commandes pour plusieurs fichiers php

- 1) `#rm /var/www/locale/monsite.pot`
- 2) `#touch /var/www/locale/monsite.pot`
- 3) `find /var/www/ -type f -iname "*.php"`
`| xgettext -j --from-code=utf-8 -f - -o /var/www/locale/monsite.pot`

Lignes de commandes pour plusieurs fichiers avec sélection d'un sous-répertoire

- 1) `rm /var/www/locale/monsite.pot`
- 2) `touch /var/www/locale/monsite.pot`
- 3) `find /var/www/ -type f -iname "*.php"`
`| grep -v /src/`
`| xgettext -j --from-code=utf-8 -f - -o /var/www/locale/monsite.pot`

Références

<https://developer.mozilla.org/en-US/docs/gettext>

<http://www.techrecite.com/gettext-tutorial-for-internationalization-and-localization-of-websites/>

Chaînes de traduction

Exemple d'une vue avec appels gettext

```
<h2><?=_('Gestion des textes des interfaces')?></h2>
<h3><?=_('Traduire')?></h3>
```

Exemple d'un fichier .po

```
#: wp-comments-post.php:29
msgid "Sorry, you must be logged in to post a comment."
msgstr "Vous devez être connecté pour rédiger un commentaire."

#: wp-comments-post.php:35
msgid "Error: please fill the required fields (name, email)."
msgstr "Erreur : veuillez remplir les champs obligatoires vides (nom, e-mail)."
http://codex.wordpress.org/Translating\_WordPress#The\_PO\_File\_Header
```

Pour inclure des variables dans une traduction

Instructions pour le traducteur

```
/// TRANSLATORS:
/// Please leave %s as it is, because it is needed by the program.
/// Thank you for contributing to this project.
printf_("La date est le %s %s %s.\n"), $jour, $mois, $annee);
```

```
xgettext --add-comments=/
```

```
#. TRANSLATORS: Please leave %s as it is, because it is needed by the program.
#. Thank you for contributing to this project.
#: src/date.php:36
```

Technique de type printf

```
printf_("La date est le %s %s %s.\n"), $jour, $mois, $annee);
```

```
msgid "My name is %s.\n"
msgstr ""
```

Le problème de cette technique est que l'ordre des variable peut être changé par le traducteur mais non respecté par printf.

Technique de type template

Une meilleure technique est d'utiliser des étiquettes de type template avec un str_replace tel que [JOUR] ou [[JOUR]] ou {{JOUR}}

```
echo str_replace(
    array(['JOUR'], ['MOIS'], ['ANNEE']),
    array($jour, $mois, $annee),
    _('Nous sommes le [JOUR] [MOIS] de l'année [ANNEE]')
)
```

```
msgid "Nous sommes le [JOUR] [MOIS] de l'année [ANNEE]"
msgstr "Welcome to the [ANNEE], [MOIS] [JOUR]"
```

Cependant, avec cette technique, il ne faut pas oublier de traduire le contenu des étiquettes s'il y a lieu, ici l'utilisation de l'initialisation du système de traduction pour les dates devrait suffire pour traduire les mois.

Éditeurs de fichiers PO	Compilateurs de fichiers MO
http://poedit.net/ http://pootle.translatehouse.org/	http://po2mo.net/ http://www.icanlocalize.com/site/

Liaison dynamique au fichier de traduction

Priorité des choix de langues

Nous procédons généralement à cette étape dans le fichier de configuration du projet. La langue choisie peut dépendre de plusieurs variables :



Programmation du lien

Programmation qui permet d'activer un langage dans le système gettext et de lier un fichier .mo

```
setlocale(LC_ALL, 'fr_CA.utf8');  
setlocale(LC_TIME, 'fr_CA.utf8');  
setlocale(LC_MESSAGES, 'fr_CA.utf8');  
setlocale(LC_MONETARY, 'fr_CA.utf8');
```

```
putenv('LANGUAGE=fr_CA.utf8');  
putenv('LC_ALL=fr_CA.utf8');
```

```
bindtextdomain("monsite", "/var/www/locale/");  
//bind_textdomain_codeset("monsite", 'UTF-8');  
textdomain ("monsite");
```

Bien sur, fr_CA doit être une variable afin de pouvoir changer la langue, ainsi que /var/www. De plus la structure du répertoire /locale/ doit être celle attendue :

```
/locale  
/en_CA  
/LC_MESSAGES  
monsite.po  
monsite.mo  
/fr_CA  
/LC_MESSAGES  
monsite.po  
monsite.mo
```

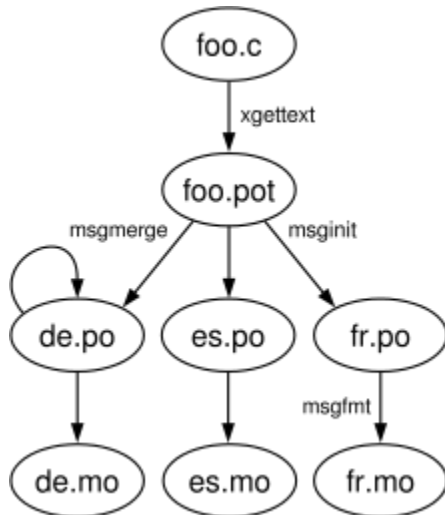
Références

<http://www.onlamp.com/pub/a/php/2002/06/13/php.html>
<http://www.onlamp.com/pub/a/php/2002/06/13/php.html?page=2>

Utilisations

Possibilités supplémentaires

Le système fournit également les commandes pour effectuer des fusions de fichier ou des extractions de PO à partir des MO et il s'applique à tout langage de programmation qui le supporte. Il est notamment très utilisé en langage C alors que le Java fournit déjà son propre système équivalent.



Source : <http://en.wikipedia.org/wiki/File:Gettext.svg>

Exemples connus

Des CMS connus utilisent GETTEXT

- http://codex.wordpress.org/Translating_WordPress
- [http://wiki.typo3.org/Localization_\(reST\)](http://wiki.typo3.org/Localization_(reST))
- <https://drupal.org/node/1189184>

Des systèmes de templates connus utilisent aussi GETTEXT

- <http://phptal.org/manual/en/split/gettext.html>
- <https://code.google.com/p/smarty-gettext/>

Des logiciels et sites web connus utilisent aussi GETTEXT

- https://wiki.openoffice.org/wiki/Localization_AOO
- http://www.phpmyadmin.net/home_page/translations.php
- <https://wiki.gnome.org/Apps/Gtranslator>