**Internationalisation (i18n) d’une application PHP avec les outils GNU « gettext »**

*Sébastien Raillard – 21 juin 2014*

Versions utilisées :

* PHP version 5.3.5 (XAMPP sous Windows 7)
* Poedit version 1.6.5 (pour Windows)

Prérequis :

* Vérifier que l’extension « php\_gettext.dll » est bien activée dans le fichier de configuration « php.ini ».
* Télécharger et installer le logiciel d’aide à la traduction Poedit compatible avec les fichiers gettext (version gratuite) ici : <http://poedit.net/>

Documentation :

* Documentation originale du site GNU : <https://www.gnu.org/software/gettext/>
* Wikipédia : <http://en.wikipedia.org/wiki/Gettext>
* Extension PHP : <http://www.php.net/manual/en/book.gettext.php>

Principe de fonctionnement :

Lors de l’écriture de chaines de caractères destinées à être affichées dans le code source, il faut utiliser certaines fonctions spéciales (« \_ », « gettext », « ngettext ») qui remplaceront éventuellement le libellé initial par sa traduction.

Il y a possibilité d’utiliser de paramètres dans les libellés et de gérer les formes singulières/plurielles.

Si les fichiers de traductions ne sont pas présents, le libellé original sera utilisé.

Il y a une notion de « domaine » dans gettext qui permet de travailler avec plusieurs jeux différents de traductions par langue.

Description des étapes :

* Ecriture dans le code des libellés avec les fonctions spéciales
* Un outil analyse l’ensemble du code source et récupère tous les libellés écrits pour en faire une liste
* Cette liste de messages originaux pourrait être enregistrée au format .pot mais cette fonction ne semble pas supportée par le logiciel Poedit (qui peut par contre importer les fichiers .pot)
* Les messages sont traduits dans une ou plusieurs langues
* Les traductions sont enregistrées dans des fichiers .po (un par langue). Le nom du fichier est important car c’est ce qui définit « domaine » dans le jargon gettext.
* Un compilateur transforme les fichiers textes .po en fichiers binaires .mo (pour accélérer la gestion des traductions)
* Les fichiers .mo sont chargé²s et les messages originaux remplacés par leurs traductions par l’extension gettext de PHP

Traduction des libellés avec Poedit : (création d’un nouveau fichier)

* Lancer l’application Poedit
* Aller dans « File » / « New »
* Choisir la langue (utiliser la version simplifiée, sans variante, comme « English » ou « French »)
* Aller dans  « File » / « Save »
* Attention, bien sauvegarder le fichier au bon endroit (cf. remarques sur l’arborescence ci-dessous) et avec le bon nom (son domaine) !
* Un nouveau fichier .po sera créé, sa version compilée .mo sera créé automatiquement lors de la sauvegarde
* Aller dans « Catalog » / « Properties »
  + Dans « Translation properties » / « Project name and version », indiquer un nom pour le projet (obligatoire)
  + Dans « Translation properties » / « Source code charset », mettre « UTF-8 »
  + Dans « Sources paths », ajouter un chemin avec comme valeur le répertoire racine de l’application racine à traduire (c’est à partir de ce répertoire que les fichiers sources seront analysés, ainsi que dans les sous-répertoires). Il faut laisser le « base path » à « . »
  + Valider les modifications avec « OK »
* Dans la barre d’outils de l’application, cliquer sur l’icône « Update », les fichiers sources seront analysés et les libellés trouvés seront listés.
* Il suffit de cliquer sur un libellé de la liste pour pouvoir inscrire sa traduction dans le champ en bas de la fenêtre
* Une fois que les traductions sont finies, enregistrer le fichier avec « File » / « Save »

Traduction des libellés avec Poedit : (modification d’un fichier existant)

* Lancer l’application Poedit
* Aller dans « File » / « Open »
* Choisir le fichier .po à modifier
* Une fois ouvert, le processus est la même que pour la création, il faut penser à faire un « Update » (bouton de la barre d’outils) pour vérifier si les libellés à traduire ont changé (ajout et/ou suppression)

Arborescence des fichiers :

L’arborescence est importante et elle doit être respectée. On peut choisir le répertoire racine, généralement appelé « locale » mais ce n’est pas une obligation. Le chemin configuré vers la racine peut être relatif ou absolu.

Voici un exemple :

* Répertoire « locale »
  + Répertoire « code langue » (il peut être sans variante, c’est-à-dire du type « en », « fr », etc.)
    - Répertoire « LC\_MESSAGES » (nom imposé)
      * Fichiers « Domaine » .mo et .po

Problèmes des fichiers .mo gardés en cache :

PHP utilise uniquement les fichiers .mo (binaires) pour des raisons de performances. La première fois qu’un fichier .mo est chargé, il est mis en cache. Du coup, si des modifications sont apportées, celles-ci ne sont pas prises en compte.

Il y a que deux moyens de faire recharger un fichier .mo modifié :

* Redémarrer le service Apache (ce qui va réinitialiser PHP et donc son cache)
* Changer le « domaine » et donc le nom du fichier .mo