Traduzioni in gino cms

Installazione

Il meccanismo delle traduzioni si basa sulle **librerie gettext**.

Se si utilizza un server Debian bisogna installare i seguenti pacchetti per poter utilizzare i comandi *xgettext* e *msgfmt*:

 # apt-get install gettext ~~php-gettext~~

gettext effettua l'upgrade dei pacchetti di **locale**. Al termine bisogna quindi riaggiornare il tutto con

 # dpkg-reconfigure locales

selezionando le opzioni necessarie.

Utility

La preparazione dei file richiede l'utilizzo delle utility **xgettext** e **msgfmt** che funzionano in questo modo:

 # xgettext hello.php (crea messages.po, occorre poi sostituire il charset appropriato)
 # msgfmt -o hello.mo messages.po (crea il file compilato hello.mo a partire da messages.po)
 # msgfmt messages.po (crea automaticamente il file compilato messages.mo)

*# xgettext --default-domain=dominio *.php* (crea dominio.po)

Per maggiori informazioni http://www.gnu.org/software/gettext/manual/

Procedura guidata

Prima di procedere, se necessario effettuare il backup del file .po

 # cd /var/www/htdocs/sito/languages/it_IT/LC_MESSAGES
 # cp messages_it.po messages_it.20120301.po

1. Creazione del file base in italiano (messages_it.po) contenente le stringhe da tradurre

 # cd /var/www/htdocs/sitoweb

*# xgettext --from-code=UTF-8 -o messages_it.po *.php lib/*.php lib/classes/*.php lib/classes/exceptions/*.php lib/classes/fields/*.php lib/classes/http/*.php lib/classes/input/*.php lib/classes/mvc/*.php lib/plugin/*.php templates/*.php test/*.php views/*.php app/attachment/*.php app/auth/*.php app/auth/views/*.php app/graphics/*.php app/graphics/views/*.php app/index/*.php app/instruments/*.php app/language/*.php app/language/views/*.php app/layout/*.php app/menu/*.php app/menu/views/*.php app/module/*.php app/page/*.php app/page/views/*.php app/phpModuleView/*.php app/searchSite/*.php app/searchSite/views/*.php app/statistics/*.php app/sysClass/*.php app/sysconf/*.php app/sysfunc/*.php*

Nel caso l'output generi un errore di directory o file not found, eliminare dall'elenco la directory indicata.

2. Effettuare una copia del file rinominandolo con il suffisso della lingua da tradurre e passare il file al traduttore

 # cp messages_it.po messages_en.po

3. Creare il file eseguibile messages.mo, uno per ogni lingua

 # msgfmt messages_it.po

il file .mo in lingua italiana deve risiedere obbligatoriamente nella directory languages/it_IT/LC_MESSAGES/. Il file in lingua inglese risiederà in languages/en_US/LC_MESSAGES/.

Per ogni lingua, oltre a creare un file eseguibile, occorre creare una directory che abbia come nome gli identificatori della lingua.

4. Reload di apache (anche se a volte funziona lo stesso)

 # /etc/init.d/apache2 reload

Aggiornare i file delle traduzioni

Può sorgere l'esigenza di aggiornare un file delle traduzioni, ad esempio a seguito dell'installazione di uno o più moduli aggiuntivi.
È importante aggiornare il file po esistente **prima** di effettuare le traduzioni.

La sintassi per aggiornare il file delle traduzioni è la seguente:

 # xgettext -j --from-code=UTF-8 -o messages_it.po app/classeNuova1/.php app/classeNuova2/*.php*

L'opzione –j permette di accorpare i file indicati nel caso in cui il file messages_it.po sia già presente.

Aggiornato il file in italiano (messages_it.po) occorre aggiornare anche i file delle traduzioni delle altre lingue:

 # msgmerge messages_en.po messages_it.po --output-file=messages_en.new.po

Naturalmente al termine di queste operazioni occorre sempre ricreare i file eseguibili.

Commenti fuzzy

Ogni volta che qualche indicazione all'interno di un file PO è incerta, in quanto predefinita o determinata automaticamente in modo non sicuro, viene aggiunto un commento speciale contenente la parola fuzzy (**#, fuzzy**). In presenza di commenti del genere si richiede un intervento manuale, dopo il quale deve essere rimossa tale parola, altrimenti **msgfmt** si rifiuta di completare la compilazione dei file PO.

Opzioni

Lista di file

Per indicare una lista di file da tradurre è possibile selezionare semplicemente *.php (tutti i file php), oppure utilizzare un file di testo:

$ xgettext --files-from=lista.txt -j --from-code=UTF-8 -o plugin.pot

Dove --files-from è un file di testo contenente la lista di file da tradurre, -j serve ad unire (join) i risultati se il file .po esiste già, --from-code indica che tipo di encoding ci si aspetta dai file in input ed infine -o il nome del file di output (dove plugin è il nome del textdomain del plugin).

Il file chiamato plugin.pot è il template su cui si baseranno le traduzioni in altri linguaggi. Create una copia e rinominatela in plugin-it_IT.po, questo sarà il file da editare, la sintassi è molto semplice e potete usare un editor di testo qualsiasi (mi raccomando che sia utf-8).

Una volta completata la traduzione dovremo compilare il file .po in modo da ottenere il nostro .mo:

$ msgfmt plugin-it_IT.po -o plugin-it_IT.mo

Charset

E’ importante individuare la codifica del file creato (ISO-8859-1 o UTF-8) ed eventualmente sostituire il charset appropriato.
Nella creazione del file è possibile utilizzare l’opzione --from-code=UTF-8 (ISO-8859-1).

Altri strumenti di gettext

Gettxet dispone di numerosi strumenti che facilitano la vita del programmatore e in particolare

gettext	estrae tutte le stringe dal codice sorgente creando un file di stringhe, ad esempio default.po
msgcat	unisce due file ad esempio file3 = file1.po + file2.po
msgmerge	fa l’intersezione di due file, prende la stringa da un file e la sua traduzione da un altro ...ad esempio file3 = file1.po U file2.po
msgremove	elimina le stringhe di contenute in un file da un secondo file. Ad esempio file3.po = file2.po - file1.po

L’estrazione di tutte le stringhe dal codice sorgente comprende anche molto testo che non è di interesse per la traduzione, come ad esempio i nomi dei font, numeri indicati come testo ecc.. Si dovrebbe evitare di includere tali stringhe nei file di traduzione e si vorrebbe che tale impostazione rimanga valida anche quando questi file vengono aggiornati. Qui è descritta una procedura che risolve questo problema automaticamente.

Gettext dispone del comando msgmkignore in grado di identificare automaticamente la maggior parte delle stringhe di servizio creandone un file, ad esempio ignore.po. Successivamente con il comando msgremove si eliminano queste stringhe dal file complessivo.

Spesso c’è anche l’esigenza di non tradurre determinate stringhe presenti nel programma, ad esempio nomi propri, comandi ecc... Benchè tali stringhe potrebbero essere ignorate in fase di traduzione, è desiderabile che siano escluse a priori dai file di traduzione.
E’ possibile aggiungere manualmente queste stringhe al file generato con msgmkignore risolvendo il problema.

Tale soluzione funziona fino a quando non si vuole generare un aggiornamento delle stringhe e quindi degli ignore. Infatti al prossimo comando msgmkignore le stringhe aggiunte manualmente vengono perse perché il file viene sovrascritto.

Una soluzione migliore è creare un proprio file di stringhe da ignorare ad esempio MyIgnore.po, da unire con msgcat al file generato da msgmkignore e successivamente utilizzare msgremove.

Gestione del plurale

```
// ....
$commenti = 1;
printf(ngettext("%d commento", "%d commenti", $commenti), $commenti);
echo "<br />";
$commenti = 10;
printf(ngettext("%d commento", "%d commenti", $commenti), $commenti);
```

Nell'esempio verrà restituita la stringa %d "commento" o %d "commenti" in base al fatto che il valore passato come terzo argomento alla funzione ngettext sia valutato come plurale o singolare in base alla regole della lingua utilizzata correntemente. La regola per decidere se un valore è da identificare come singolare o plurale viene specificata nell'header del file .po utilizzando la seguente sintassi:

```
Plural-Forms: nplurals=2; plural=n != 1;
```