Internationalisatie en localisatie van shell scripts

Presentatie

Doel, reikwijdte en doelgroep

Dit document is bedoeld om ontwikkelaars, maintainers en vertalers te helpen bij het schrijven, onderhouden en vertalen van ge-internationaliseerde en dan gelocaliseerde shell scripts, met gebruikmaking van de gereedschappen die door GNU gettext worden geleverd.

Het referentiedocument is de handleiding getiteld GNU `gettext' utilities.

De handleiding behandelt alle programmeertalen die bruikbaar zijn met gettext, met een speciale focus op de taal 'C'.

De POSIX specificatie is aanbevolen leesvoer, met name de volumes Basis Definities en Shell en Gereedschappen.

In contrast met de handleiding blijft de reikwijdte van het onderhavige document beperkt tot shell scripts.

Theorie van de operatie

Het doel is om berichten (gewoonlijk stukken tekst) die de uitvoer zijn van shell scripts, weer te geven op het systeem van de gebruiker in diens voorkeurstaal.

De gebruiker geeft zijn/haar voorkeur aan door het instellen van de LANG of de LANGUAGE omgevingsvariabele (de laatste bevat een prioriteitenlijst van talen die voor het weergeven van berichten gebruikt mogen worden).

Het **Internationalisatie** proces (verkort tot I18N) bestaat uit:

- de teksten markeren in de shell scripts die de te vertalen uitvoer-berichten vormen,
- vervolgende de gettext gereedschappen inzetten om uit deze verzameling gemarkeerde scripts een *berichten sjabloon-catalogus* te vormen.

Een berichten sjabloon-catalogus wordt gewoonlijk een “Portable Object Template” of POT bestand genoemd.

Een POT bestand, leesbare platte tekst, bestaat voornamelijk uit de geëxtraheerde tekst-reeksen, vooraf gegaan door “msgid” wat zoveel betekent als “message identifier”. Bij elke “msgid” tekst behoort een vertaling van dat bericht, en deze tekst-reeks wordt vooraf gegaan door het woord “msgstr” wat staat voor “message string”.

Het **Localisatie** proces (verkort tot L10N) bestaat uit:

- het genereren van aparte “Portable Object” of PO bestanden voor iedere doel-taal uit het enkele POT bestand,
- alle “msgstr” teksten voorzien van vertalingen in ieder PO bestand,

- deze PO bestanden controleren/verifiëren na de vertaalslag,
- ieder PO bestand individueel compileren tot een “Machine Object” of MO bestand.

De MO bestanden die bedoeld zijn voor de machine, niet de mens (vandaar de naam) worden traditioneel opgeslagen als:

```
/usr/share/locale/<locale>/LC_MESSAGES/<software naam>.mo
```

In het bovenstaande pad is <locale> een *locale* code in de vorm <tt[_LL]>, waar tt de twee-letterige code is van de doel-taal zoals gedefinieerd in de ISO 639-1 standaard, en LL (indien aanwezig) is de twee-letterige land-code van deze locale zoals gedefinieerd in ISO 3166.

Ieder gemarkeerd script moet het volgende commando bevatten:

```
export TEXTDOMAIN=<software naam>
```

Tijdens het uitvoeren van het script wordt zo 'gettext' in staat gesteld om het juiste MO bestand te vinden en iedere gemarkeerde bericht-tekst weer te geven in de voorkeurstaal zoals die door de LANG of LANGUAGE omgevings-variabele is bepaald.

Procesdiagrammen

Laten we aannemen dat een bepaalde software bestaat uit een verzameling shell scripts die we willen internationaliseren en localiseren.

De onderstaande diagrammen tonen een overzicht van iedere fase van de betrokken processen: internationalisatie, localisatie, gebruik en onderhoud.

Deze diagrammen zijn hybride, dwz ze tonen data zowel als acties.

Onder deze acties vallen de uitvoering van een aantal programma's uit de `gettext` verzameling:

- `gettetxt`: markeert de te internationaliseren tekstreeksen, en toont later de gelocaliseerde berichten tijdens de uitvoering van het script
- `xgettext`: extraheert de gemarkeerde tekstreeksen uit een verzameling shell scripts om er een *POT* of een *PO* bestand van op te bouwen
- `msgcmp`: controleert een PO bestand op consistentie aan de hand van een ander PO of POT bestand
- `msginit`: schrijft een PO bestand met een POT bestand als de input
- `msgfmt`: genereert een MO bestand met een POT bestand als de input
- `msgmerge`: samenvoegen of wijzigen van PO of POT bestanden

In onderstaande diagrammen worden de `gettext` programma's omgeven door vierkante haken.

(1) Internationalisatie

```
Verzameling shell scripts ———> Preparatie ———> Gemarkeerde shell scripts
```

```
┐
│
          ┌──────────────<── software.pot <─── [xgettext]
<─────────────────────┘
          │
          ├─────> (2) Localisatie
          │
          └─────> (4) Onderhoud
```

(2) Localisatie (voorbeeld met Frans en Nederlands als talen).

```
     ┌──────────────<── software.pot <── (1) Internationalisatie
     │
     │
     ├──> [msginit] ──> fr_FR.po ──> PO editor ──> fr_FR.po ──> [msgcmp] ──┐
     │                                                                      │
     │    ┌─ installatie <── fr_FR.mo <── [msgfmt] <─┬─ fr_FR.po controle <─┘
     │    │                                          │
     │    │                                          └─────> (4) Onderhoud
     │    │
     │    └──> /usr/share/locale/fr_FR/LC_MESSAGES/software.mo —> (3) Gebruik
     │
     └──> [msginit] ──> nl_NL.po ──> PO editor ──> nl_NL.po ──> [msgcmp] ──┐
                                                                           │
         ┌─ installatie <── nl_NL.mo <── [msgfmt] <─┬─ nl_NL.po controle <─┘
         │                                          │
         │                                          └─────> (4) Onderhoud
         │
         └──> /usr/share/locale/nl_NL/LC_MESSAGES/software.mo —> (3)
Gebruik
```

(3) Gebruik

Veronderstel dat een van de scripts, “`myscript.sh`” het volgende commando bevat:

```
gettext "Good morning"
```

en dat “Good morning” als volgt vertaald is in de berichten catalogi:

```
/usr/share/locale/fr_FR/LC_MESSAGES/PACKAGE.mo —> "Bonjour"
/usr/share/locale/nl/LC_MESSAGES/PACKAGE.mo —> "Goedemorgen"
```

Hier is wat de gebruiker zal zien afhankelijk van de instelling van de LANG omgevingsvariabele:

```
             ┌──────────────<── (2) Localisatie
             │
LANG=fr_FR ──├─────> sh myscript.sh of ./myscript.sh ──────> "Bonjour"
             │
```

```
 LANG=nl_NL └──> sh myscript.sh of ./myscript.sh ──> "Goedemorgen"
```

(4) Onderhoud

Het onderhoudsproces kan worden gestart door het creëren, wijzigen of verwijderen van een script.

In onderstaand diagram moet het onderdeel van het proces dat begint met het 'msmerge' commando herhaald worden voor ieder beschikbaar PO bestand.

Het is daarom aan te raden om een actuele lijst bij te houden van alle beschikbare vertalingen in de vorm van PO bestanden.

```
      Shell scripts gewijzigd en gemarkeerd ──> [xgettext] ──>
software.pot ──┐
│
                                    (1) Localisatie ──> <locale>.po
──>┬<──────┘
                                                                  │
┌─ [msgcmp] <── <locale>.po <─ PO editor <── <locale>.po <─ [msmerge] <─┘
│
└──> <locale>.po controle ─> [msgfmt] ─> <locale>.mo ─> installatie  ──┐
                                                                       │
                  /usr/share/locale/<locale>/LC_MESSAGES/software.mo <──┘
```

Het onderhoudsproces kan ook gestart worden door een wijziging van een berichten catalogus voor een specifieke taal (om bijvoorbeeld een fout te corrigeren).

Deze variant van het proces is korter:

```
┌─ [msgcmp] <── <locale>.po <── PO editor <── <locale>.po <── Wijziging
gewenst
│
└──> <locale>.po controle ──> [msgfmt] ──> <locale>.mo ──> installatie  ──┐
                                                                          │
                   /usr/share/locale/<locale>/LC_MESSAGES/software.mo <──┘
```

Internationalisatie proces

Deze paragraaf is bedoeld voor ontwikkelaars en maintainers.

Het internationalisatie proces omvat de volgende taken:

1. Bereid scripts voor op internationalisatie
2. Markeer de tekstberichten die gelocalizeerd moeten worden
3. Gebruik 'xgettext' om een sjabloon catalogus van de tekstberichten te produceren

Scripts ter internationalisatie voorbereiden

Deze taak is nodig voor shell scripts die nog niet aan de voorwaarden voldoen voor internationalisatie.

Technische notitie: Gettext vereisten voor shell scripts.

Onderstaande opsomming van vereisten is niet compleet.

Het noemt enkel de voornaamste die de ontwikkelaar of maintainer aangeraden worden, gebaseerd op de auteur's eigen ervaringen.

Gettext vervangt tijdens de uitvoering van het script (dwz 'runtime') tekstberichten die bijvoorbeeld worden uitgestuurd door:

- Een “echo” commando of
- een programma (zoals '`dialog`' bijvoorbeeld)

met vertaalde tekst (die gevonden wordt in de berichten catalogus voor de taal die middels $LANG of $LANGUAGE ingesteld is)

Maar deze vervanging vindt alleen plaats wanneer aan de volgende voorwaarden voldaan is:

- Een MO bestand is aanwezig in het pad dat samengesteld wordt uit de TEXTDOMAIN omgevingsvariabele tot <directorynaam>/<locale>/LC_MESSAGES/text_domain.mo.
  Om een voorbeeld te geven, als TEXTDOMAIN=software en $LANG=de_DE.utf8, dan zal `gettext` zoeken naar: <directorynaam>/de_DE/LC_MESSAGES/software.mo
  <directorynaam> kan worden ingesteld door de waarde van de TEXTDOMAINDIR omgevingsvariabele, en anders wordt een standaardwaarde gebruikt.
  In Slackware Linux bijvoornbeeld is de standaard waarde `/usr/share/locale`.
  Er zijn ook alternatieve lokaties: bijvoorbeeld als <locale> gelijk is aan “de_DE” dan kan het MO bestand ook worden geplaatst in <directorynaam>/de/LC_MESSAGES/ in plaats van <directorynaam>/de_DE/LC_MESSAGES/
- TEXTDOMAIN variabele wordt geëxporteerd, voordat enig *`gettext` commando wordt uitgevoerd.
- `gettext.sh`, dat de functies `eval_gettext` en `eval_ngettext` aanbiedt, wordt ge'*source*'d voor de eerste aanroep van een van deze functies.
- Een “msgid” tekstreeks in het MO bestand past exact bij het argument voor `gettext` (of `eval_gettext` als de tekstreeks een parameter expansie bevat).
- De corresponderende “msgstr” tekstreeks bevat geen backslash die gevolgd wordt door een spatie.
- De “msgstr” tekstreeks string begint en eindigt met een newline - of niet - precies zoals de “msgid” het doet.
- Indien de tekstreeks een parameter expansie bevat, moet `eval_gettext` worden gebruikt in plaats van `gettext`.
- “De variabele namen moeten enkel bestaan uit alfanumerieke of underscore ASCII karakters, niet starten met een cijfer en niet leeg zijn; anders wordt zo een variabele referentie genegeerd.” (gettext handleiding)
- Parameter expansies worden escaped met een enkele backslash zoals dit:
  \$parameter of \${parameter}
  tenzij de `eval_gettext` aanroep zich bevindt binnen een commando substitutie zoals dit:

“`eval_gettext ”…“`” of “$(eval_gettext ”…“)“
In het laatste geval zijn drie backslashes noodzakelijk zoals dit:
\\\$parameter of \\\${parameter}.
- Enkel de voorkomens $parameter en ${parameter} van een parameter expansie worden gebruikt binnen een `eval_gettext` argument (alle andere voorkomens zijn verboden).
- Positionele parameters, speciale parameters en commando substituties worden *niet* gebruikt binnen een argument van `gettext` of `eval_gettext`.

Een praktisch gevolg van met name de twee laatste vereisten, is het advies om alle positionele parameters, speciale parameters, commando substituties en niet-toegestane vormen van parameter substituties aan de “upstream” ontwikkelaars te melden om een variabele-naam toe te laten wijzen, en dan geëxpandeerd in de tekstreeks die als argument van `eval_gettext` of `eval_negettext` dient.

Tip: als een tekstreeks aanwezig is als een “msgid” in een berichten catalogus en toegewezen aan een variabele-naam in een script, dan zullen de commando's: ”`gettext $parameter`” en “`gettext ${parameter}`” de vertaalde tekst uitgeven tijdens de script-executie (runtime), zelfs wanneer '`xgettext`' dat commando zo negeren bij het verwerken van het script, omdat '`gettext`' gebruikt wordt in plaats van '`eval_gettext`'. Dit kan handig zijn. In dat geval moet de parameter expansie niet ge-escaped worden.

Markeer berichten om te lokaliseren

Aanbevolen wordt om de volgende tekstberichten te markeren:

- argumenten van een niet omgeleid '`echo`' commando.
- argumenten van omgeleide '`echo`' commando's indien in latere instantie de tekst toch op het scherm van de gebruiker wordt afgebeeld.
- argumenten van andere commando's die het tekstbericht tonen, bijvoorbeeld het '`dialog`' programma.

In tegenstelling daarmee is het aan te raden om de volgende niet te markeren:

- commentaren bedoeld voor lezers van het script,
- tekstreeks wiens waarde later zal worden verwerkt, bijvoorbeeld als argumenten voor een '`case`' samengesteld commando, of <tag> argumenten van een '`dialog –menu`' commando.

Soms schrijft het shell script andere (nieuwe) shell scripts.

Dan zal de ontwikkelaar of maintainer van geval tot geval moeten beslissen wat er gemarkeerd moet worden, afhankelijk van de reikwijdte van de voorgenomen internationalisatie.

Gebruik 'xgettext' om een sjabloon-catalogus van berichten te produceren

Een keuze moet worden gemaakt om een enkel POT bestand te maken voor de software als geheel, danwel een apart POT bestand voor verschillende verzamelingen scripts. Overwegingen zullen zijn, welke keuze minimaliseert het onderhoudswerk, hoe kan het werk aan de localisaties het beste worden geörganiseerd, wat is de relatieve frequentie van de wijzigingen op de verschillende

verzamelingen van scripts die samen de software vormen, en de relevantie van de onderscheidende features zoals setup versus configuratie versus pakketbeheer.

De auteur produceert gewoonlijk een enkel POT bestand, maar iedereen moet zijn eigen afwegingen maken.

indien de software bestaat uit een veelvoud van scripts die op verschillende plaatsen staan of opgenomen zijn in verschillende pakketten, kan het voordelig zijn om een kopie van alle scripts te verzamelen in een enkele directory, en/of in een apart tekstbestand al deze scripts te noteren inclusief hun padnamen.

Het POT bestand wordt gegenereerd met het 'xgettext' commando (zie de handleiding of 'xgettext –helpxgettext –help' voor de details).

Neem de volgende opties mee in het commando:

```
-L Shell (vanzelfsprekend!)
--strict (ter ondersteuning van controles en beheer van de berichten
catalogi)
-c       (om commentaren in het POT bestand op te nemen die nuttig zijn voor
de vertalers)
-n       (identificeert het bronbestand en het regelnummer van ieder
tekstbericht.
         Dit is de standaard.)
```

Wanneer het POT bestand eenmaal gegenereerd is moet je controleren dat het secties bevat voor alle *gettext aanroepen in de shell script(s).

Localisatie proces

Wanneer het POT bestand klaar is, genereert het 'msginit' commando daaruit een PO bestand voor iedere doel-taal.

In PO bestanden mogen de “msgid” tekstreeksen nooit gewijzigd worden, omdat er anders helemaal geen vertaling zal plaatsvinden tijdens de uitvoering van het script (runtime).

Met het 'msgcmp' commando kun je na de vertaalslag ieder PO bestand controleren tegen het POT bestand, om je te verzekeren dat alle tekstberichten vertaald zijn.

De vertaler kan het 'msgfmt' commando gebruiken om de layout van de vertaalde tekst te verifiëren.

Het PO bestand moet zorgvuldig bewaard worden, omdat dit opnieuw nodig is wanneer er onderhoud gepleegd wordt op de vertaling (het is mogelijk om met het 'msgunfmt' commando uit een MO bestand een PO bestand te re-creëren maar daarbij gaat wel de context verloren, waardoor het resultaat nagenoeg onbruikbaar is).

Het gecontroleerde PO bestand wordt overgedragen aan de maintainer, die dan 'msgfmt' gebruikt om het MO bestand te produceren en te installeren.

Gebruik

Het enige dat de gebruiker hoeft te doen is het instellen van zijn of haar voorkeurstaal of -talen.

De standaard manier om dat voor elkaar te krijgen is door de LANG omgevingsvariabele de juiste waarde te geven.

Dit kan tijdens de script-uitvoering (runtime) gedaan worden door het commando dat wordt gebruikt om het script uit te voeren vooraf te laten gaan door de tekst LANG=<locale>, maar meestal zal de gebruiker zijn voorkeurstaal liever permanent willen instellen.

In Slackware Linux bijvoorbeeld, wordt dat bereikt door het aanpassen van het relevante bestand `/etc/profile.d/lang.sh` (voor Bash-achtige shell) en/of `/etc/profile.d/lang.csh` (voor csh-achtige shell). De inhoud van deze bestanden wijst zichzelf door de er in opgenomen commentaren.

De wijzigingen in die bestanden worden pas effectief bij de eerstvolgende login (bijvoorbeeld na een reboot).

Het is aan te raden om een W UTF-8 locale te gebruiken.

Voor een meertalige gebruiker (polyglot) is de gettext-specifieke LANGUAGE omgevingsvariabele een mogelijkheid om een geprioritiseerde lijst van talen te specificeren.

Indien bijvoorbeeld LANGUAGE wordt ingesteld op 'de:fr' dan zal een duitse vertaling worden gebruikt indien aanwezig, en anders wordt een franse vertaling worden gebruikt indien aanwezig, en in het uiterste geval zullen tekstberichten worden getoond in de originele taal, gewoonlijk engels. Zie de gettext handleiding voor details.

Onderhoud

Meestal zal het onderhoudsproces worden gestart door het creëren, wijzigen of verwijderen van een script.

In zo'n geval zal de maintainer een nieuw POT bestand genereren met behulp van '`xgettext`' en dat dan doorspelen naar de vertalers.

De vertalers gebruiken dan het nieuwe POT bestand om hun eigen respectieve (eerder opgeslagen) PO bestanden te verversen met het '`msgmerge –update`' commando.

Met een PO editor worden de vertalingen bijgewerkt en eventueel opnieuw compleet gemaakt, met een focus op nog onvertaalde berichten en ook diegene die als “fuzzy” (onduidelijk/vaag) worden aangemerkt in de PO bestanden.

Na afloop wordt het PO bestand gecontroleerd tegen het POT bestand met '`msgcmp`', zorgvuldig opgeslagen, en overgedragen aan de maintainer. Die zal het nieuwe MO bestand genereren met '`msgfmt`' en dit bestand installeren op de juiste plek zoals gedefinieerd in het initiële localisatie proces.

Het onderhoudsproces dat wordt gestart omdat een PO bestand van een specifieke taal een wijziging

behoeft, is vergelijkbaar maar korter: het start met een wijziging van het relevante PO bestand door de vertaler. Om de werklast voor de maintainer te beperken zou deze van zijn vertalers kunnen eisen dat hij alleen maar complete en goed gecontroleerde vertalingen toegestuurd krijgt.

Practische aanbevelingen voor ontwikkelaars en maintainers

Many English words are polysemous: their meaning can only be determined from the context of their usage. As a practical consequence, the more context you provide, the more accurate the translation can be.

Example: recently, while downloading a software I saw something like this:
31min gauche
Go figure? After a while I realized that “left” had been translated “gauche” (as in “left hand”).

Also, order of words in a sentence vary upon language, furthermore not all languages are written left to right. Thus, mark entire paragraphs, or at least entire sentences, not lines, let alone isolated words but in special cases.

For instance, if text paragraphs were split in lines displayed by 'echo' commands, replace all consecutive 'echo' commands by a single 'gettext' or 'eval_gettext' command.

Do not fear to include the variable substitutions in the sentences, PO editor will check that they be present as is in the translations.

Aanbevelingen voor 'dialog' programma.

The 'dialog' program provides an UI taking the form of dialog boxes.

There are other programs with similar feature, to which I guess (only a guess), these recommendations are also applicable.

Bear in mind following considerations, when making or reviewing the design choices for dialog's boxes.

- Messages translated in other languages will often be significantly longer than the original (usually in English) ones.
- In situations where only VGA drivers are available (e.g. in text installers) screen display is generally restricted to 25 rows of 80 columns with most widely used fonts, but in practice word wrapping can occur if line's length is more than 74 characters.
  As a consequence, for static layouts text lines' length should be at most 74 characters.
- Vertical scrolling of text is widely accepted as frequently used to display web pages, and sometimes unavoidable.
  On the contrary, horizontal scrolling should be avoided as much as possible.

Therefore I suggest to:

- renounce to tightly adjust the dimensions of the boxes to the size of English text as the translation will probably break your carefully crafted layout, unless you impose unreasonable (IMO) constraints to the translators,
- in particular, not narrow boxes' width to what is strictly needed for displaying English texts,

especially in tabular layouts where the text can't flow on next lines,
- favor a fluid layout of the displayed text over a fixed one to avoid too long lines in translations, whose complete display would then necessitate horizontal scrolling (which, moreover, is not always possible).

In particular, I recommend to favor options which take as first argument a text string instead of a file, to allow line wrapping. It is still possible to preserve the intended layout using white spaces for indentation.

For instance,
dialog <common-options> –textbox <file> <height> <width>
can be replaced with
dialog –no-collapse <common-options> –msgbox "`cat <file>`" <height> <width>

Practische aanbevelingen voor vertalers

Afhankelijk van de hoeveelheid werk en beschikbare krachten, kan er een enkele vertaler of een groep vertalers zijn per doel-taal. In alle gevallen is het aan te bevelen dat minstens één persoon verantwoordelijk wordt voor de organisatie van het werk in het team, voor de controles van de vertalingen en het opsturen van het gecontroleerde PO bestand naar de maintainer(s). We zullen deze persoon de team coordinator noemen.

Voel je niet verplicht om een letterlijke vertaling op te leveren. Niet alleen is dit zelden de beste manier om de bedoelingen over te dragen, maar het leidt ook vaak tot zinnen die te lang zijn om in de toegestane ruimte te passen.

Gebruik een editor die toegespitst is op het werken met PO bestanden, en niet een 'generieke' tekstverwerker. Dit zal enerzijds verhinderen dat er per ongeluk de 'msgid' tekstreeksen zelf worden vertaald en anderzijds ook het vertaalwerk en additionele controles vergemakkelijken; zoals de aanwezigheid van een variabele in de vertaling met dezelfde spelling als in het origineel.

Kies tijdens het vertalen voor een font met een schreef en vaste breedte (een "monospaced" of typmachine font), zoals Courier. Met zo'n font kun je karakters van elkaar onderscheiden die anders (bijna) hetzelfde zouden lijken, en regellengtes controleren wanneer dat belangrijk is.

Houd de indeling van de tekstberichten in de gaten. Vergelijk deze met de context in het betreffende bronbestand. Wat nog beter werkt is gewoon uitvoeren van het vertaalde script en het resulterende tekst controleren.
Dit is vooral van belang bij het vertalen van dialoog vensters. In het bijzonder moet vermeden worden dat lange zinnen op een enkele regel blijven staan, wanneer het duidelijk is dat de tekst niet kan overvloeien naar de volgende regel.

Houd in gedachten dat bij het werken in VGA mode (zoals in de tekst installers in het bijzonder) de regelbreedte in principe gelimiteerd is tot 80 karakters. In praktijk echter vaak niet meer dan 74.

Voeg geen vraagtekens toe als die niet aanwezig zijn in de oorspronkelijke tekst.

Indien de tekst refereert aan labels (tekst op de knoppen) van dialoog vensters zoals "OK", "Yes", "NO", "Continue", "Cancel", zul je moeten controleren hoe deze labels in de dialoog interface vertaald

worden in je eigen taal. Gebruik dezelfde woorden.

Vermijd informeel taalgebruik en complex technisch woordgebruik.

Om een regel af te snijden in een dialoog venster gebruik je de eind-tekst `\n`. Het indrukken van de Enter toets zal *geen* “nieuwe regel” karakter invoegen in de tekst die de lezer ziet.

Tot slot moet aan de vereisten worden voldaan die bij het gebruik van `gettext` horen:

- Indien een woord voorkomt in de originele tekst dat begint met een dollar-teken moet het ook in de vertaalde tekst aanwezig zijn met precies dezelfde spelling (met inbegrip van hoofd/kleine letters).
- De vertaalde tekst moet een “nieuwe regel” karakter (ook wel “line feed” genoemd, gerepresenteerd door “\n”) aan het begin of aan het eind hebben staan als de oorspronkelijke tekst dat ook heeft. Omgekeerd, als de oorspronkelijke tekst dit karakter niet heeft, dan moet de vertaalde tekst het ook niet bevatten.
- Een enkel backslash karakter “\” (omgekeerde schuine streep) is niet toegestaan in de vertaling.

Om je vertaling te controleren tegen de `gettext` vereisten kun je het volgende commando uitvoeren:

```
msgfmt -c <naam van het PO bestand>
```

Waarschuwing over vertaling van man paginas

Behoud zorgvuldig de syntax van man pagina's zoals die in de Engelse markup gevonden wordt. Wat je bijvoorbeeld niet moet vervangen:

- 'B<' door 'B <' (voeg geen spatie toe)
- 'B<' door 'b<' (houd B als een hoofdletter - en vervang het ook niet door de Griekse hoofdletter BETA die er op het scherm identiek uitziet)
- “I” door '|' (vervang de hoofdletter I niet door een pipe symbool)

Bewaar bij een vertaling van shell commando's de Engelse pad-namen waar nodig. Maar je mag (en moet) argumenten vertalen die bij uitvoering van het commando vervangen worden door een waarde of string - zoals 'packagename'.

Bronnen

- Origineel geschreven door Didier Spaier
- Vertaling door Eric Hameleers

howtos, gettext, shell, scripts, internationalization, localization, i18n, l10n, author alienbob