Internationalisation & Localisation

Luk Claes

Debian Developer
http://people.debian.org/~luk
luk@debian.org

2005 June 23th

Overview

i18n & l10n

i18n internationalisation:

- to make a program ready for localisations
- done by the programmers and/or maintainers

l10n localisation:

- to translate and to adapt for cultural conventions
- done by the translators

Localisation

charset	see locale.gen
LC_CTYPE	upper, lower, alpha, digit, space, cntrl, etc.
LC_MONETARY	symbol, decimal point, thousands seperator, etc.
LC_NUMERIC	decimal point, thousands seperator, grouping
LC_TIME	names and abbreviations of weekdays and months, date and time format
...	more info in locale(5)

Overview

Preparing Program Sources

- Make localisation functions known to all modules needing them
- Properly trigger gettext operations at initialisation of the program
- Identify and mark all constant strings needing translation

Preparing Program Sources (I)

Make l10n functions known to all modules needing them.

- modules having constant string needing translation
- modules containing output calls with a format string that could be translated

```
#include <libintl.h>
#define _(string) gettext (string)
```

Preparing Program Sources (II)

Properly trigger gettext operations at initialisation of the program.

- main function

```
#include <locale.h> // For LC_ALL

setlocale (LC_ALL, "");
bindtextdomain (PACKAGE, LOCALEDIR);
textdomain (PACKAGE);
// PACKAGE is normally the name of the package
// LOCALEDIR is normally /usr/share/locale
```

Preparing Program Sources (III)

Identify and mark all constant strings needing translation.

- Decent English style: no slang, abbreviations, ambiguity, etc.
- Entire sentences: to avoid grammatical problems
- Split at paragraphs: to spot changes quickly

Just put the constant strings needing translation between "_(" and ")"

Making the translation template (POT file)

- xgettext –keyword=_ -o $(PACKAGE).pot -f POTFILES.in

Overview

Start Translating

- Copy the POT file to XX.po where XX is the ISO code of your language
- Edit the PO file using kbabel, gtranslator, poedit or . . .

Structure of a PO file

- Header:
  - Project-Id-Version: PACKAGE VERSION
  - POT-Creation-Date: ...
  - PO-Revision-Date: ...
  - Last-Translator: NAME <EMAIL>
  - Language-Team: NAME <MAILING_LIST>
  - charset=CHARSET
  - ...
- Messages:
  - WHITESPACE
  - # Translator's comments
  - #. Automatic comments (from the source files)
  - #: FILE:LINE_NUMBER
  - #, FLAG (fuzzy, c-format, no-c-format, ...)
  - msgid "Original string"
  - msgstr "Translation"

Changing POT file

- xgettext –keyword=_ -o $(PACKAGE).pot -f POTFILES.in
- msgmerge -U XX.po $(PACKAGE).pot

Installing translations

- msgfmt -c -o XX.mo XX.po
- mkdir -p $(LOCALEDIR)/XX/LC_MESSAGES
- cp XX.mo $(LOCALEDIR)/XX/LC_MESSAGES/$(PACKAGE).mo

Using translations

- LANG=XX /usr/bin/program

Note that LC_ALL has precedence on LC_<whatever> and LANG.

Acknowledgements

- Gettext manual
- Gettext tutorial
- several manpages

Overview

Converting debconf templates

- debconf-gettextize debian/<templates>
  - Creates debian/po/xx.po
  - Creates debian/po/POTFILES.in
- check that new templates have (only) _ for translateable fields
- remove obsolete files
- build depend on debhelper or po-debconf
- adjust debian/rules
- fix encoding of debian/po/*.po.unknown

Update templates

- debconf-updatepo (in clean target)

Note that comments are treated as comments for translators.

Install translations

- dh_installdebconf or:
  - po2debconf debian/templates > debian/templates.merged
  - cp debian/templates.merged debian/tmp/DEBIAN/templates

Overview

Mailing lists

- debian-i18n@lists.debian.org
  - Call for translations
  - Topics interesting for all translators
  - d-i translation coordination
  - Internationalisation issues
- debian-l10n-<lang>@lists.debian.org
  - Requests for translation to English
  - Translation coordination

Translation status

- http://www.d.o/intl/l10n/po-debconf/xx
- http://www.d.o/intl/l10n/po/xx
- http://people.d.o/ seppy/d-i/translation-status.html
- http://www.d.o/devel/website/stats/xx
- ...
- Translation coordination framework

Translation coordination framework

A bot interprets email subjects and shows a status (dl10n on Alioth), an example is http://dutch.debian.net

[ITT] Intention To Translate

[RFR] Request For Review

[LCFC] Last Chance For Comments

[BTS] link to bugnumber

[DONE] translation is incorporated

Note that this framework supposes a review process as a quality assurance measure. Note also that there are some recent discussions and proposals on debian-i18n@l.d.o and http://wiki.debian.net/?l10nCoordination

How to help translating in Debian?

- Contact debian-l10n-<lang>@l.d.o
- Look at http://www.d.o/intl/<Lang> (with and without capital)
- For new languages or when there is no debian-l10n-<lang>@l.d.o, contact debian-i18n@l.d.o

Thank you

- Thank you for your attention!
- Any questions?
- Luk Claes <luk@debian.org>
  - Debian l10n Dutch
  - see http://people.debian.org/~luk for more