i18n in Debian

A look at projects and tools
http://people.debian.org/~jfs/debconf/i18n/

Javier Fernández-Sanguino Peña
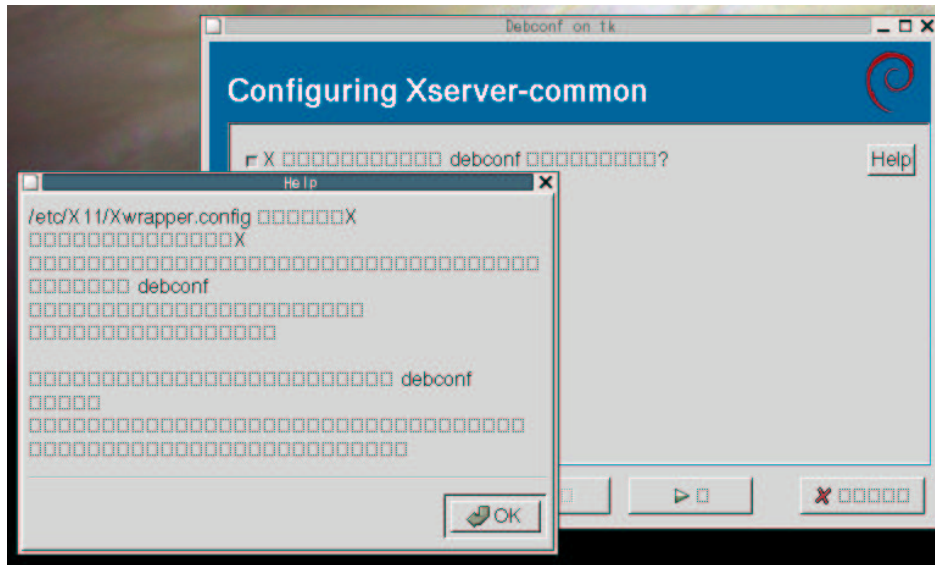
`jfs@debian.org`

i18n projects in Debian

Goals of this conference

- Show why i18n is important in Debian (isn't it obvious?)

- Show the current projects regarding i18n/l10n in Debian.

- Discuss on the issues that people that translate currently encounter.

- Make some proposals.

i18n in Debian: What's the rush?

Mojibake = Not nice.

It really means unintelligible gibberish displayed on a screen or printed on a paper when a software fails to handle encodings or fonts in the data stream.
http://www.debian.or.jp/ kubota/mojibake/

Why is i18n/l10n important?

- If we want Debian to be a *Universal* OS system it has to be usable by users respectless of:
  - the language they use
  - the way that language is represented in the computer

- Users need to be able to use it but also to
  - Read about it (documentation)
  - Ask others about it (mailing lists)

What is our status?

- We provide i18n installation of 21 langs in d-i (only 10 >5 po translated), 23 in boot-floppies.

- Release docs: 15 languages (only fully translated in 9 langs)

- Documents: 20 languages (but only 6 with more >5/37)

- Specific work in some subprojects: Debian-jp!

- Statistics: http://www.debian.org/intl/l10n/

What is other's status?

- RedHat provides an i18n installation, web/documents is not translated and few system tools are.

- Mandrake provides i18n installation/tools (60+ langs?) but web/documents are not fully translated (5 langs and not all available).

- OpenSBD 15 langs in webpages (only web pages atm)

- FreeBSD 4 langs in webpages (other translations of some docs)

Debian is probably the most *i18nized* free software OS.

Current i18n/l10n projects

There are a number of projects:

- Translation of documentation (DDP+manpages)

- Translation of web server content

- l10/i18n of Debian-specific tools:
  - d-i
  - dselect
  - dpkg
  - apt
  - debconf

- l10/i18n of non-Debian specific programs.

- Translation of packages descriptions (DDTP)

Where does discussion take place?

- Coordination among developers/translators for i18n is done at debian-i18n@l.d.o

- debian-l10n-XXX@l.d.o mailing lists to provide translators with discussion forums

- debian-user-XXXX@l.d.o mailing lists to provide users language-specific forums

Tools translators have to use

- CVS

- Gettext

- Mail management systems (BTS, DDTP)

- Custom tools (usually scripts) to verify "up-to-date"ness (for wml, for manpages)

- Po-management tools (kbabel, gtranslator) for *personal* use

Things developers need to know

- encoding issues
  (http://www.debian.org/doc/manuals/intro-i18n)

- gettext

- po-debiandoc

- po-debconf

- poxml (KDE)

- ...

Issues in i18n/l10n'ing Debian: organisation

- Translation coordinators have no real power (unless if DD) ("Do not touch l10n files"! thread in -devel/-i18n)

- Multiple interfaces for the same tasks.

- How can I see which are the current bugs related to lang X?

- Translations are not always distributed properly, usefulness drops (#64278, #172482)

Issues in i18n/l10n'ing Debian: technical

- How best to configure the system to use lang X?

- How to check up-to-date status of content: warn users and developers.

- Encoding issues (remember mojibake!)
  - Wrong Assumptions: "one character is one byte", "words are separated by whitespaces"
  - Use unicode
  - Translation is a small topic in i18n area

- How should l10n stuff be distributed? (separate packages or a single package?)

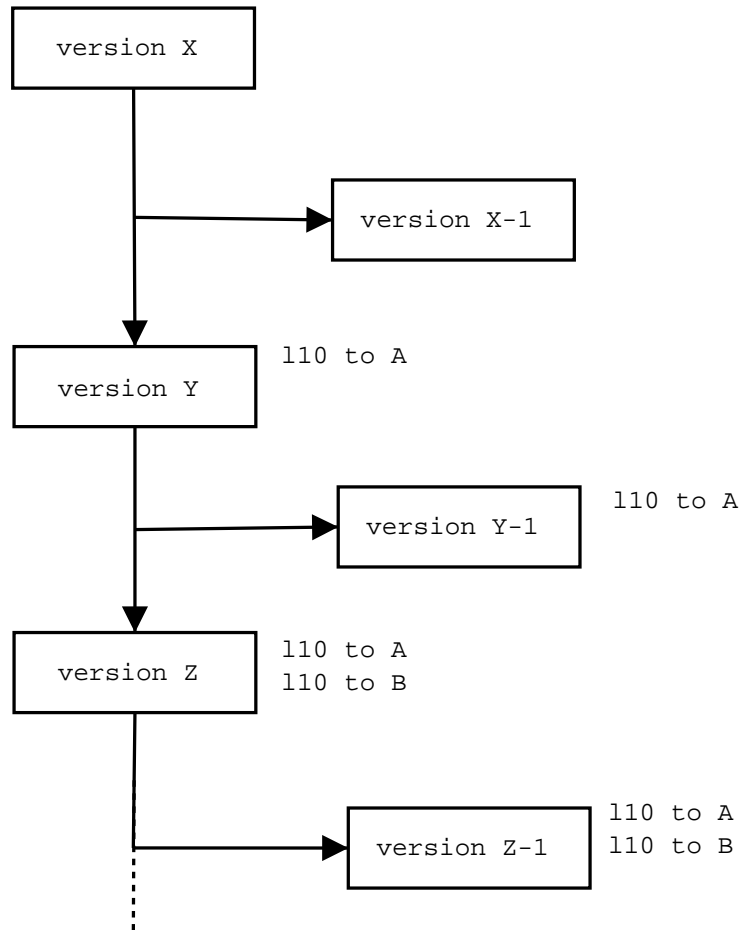# Issues in i18n/l10n'ing programs not part of Debian

There are two possible approaches:

- Leave i18n/l10n up to upstream

- Do it in Debian and then contribute to upstream

Both have issues (effort duplication, conflicts, divergence, up-to-date checks..)

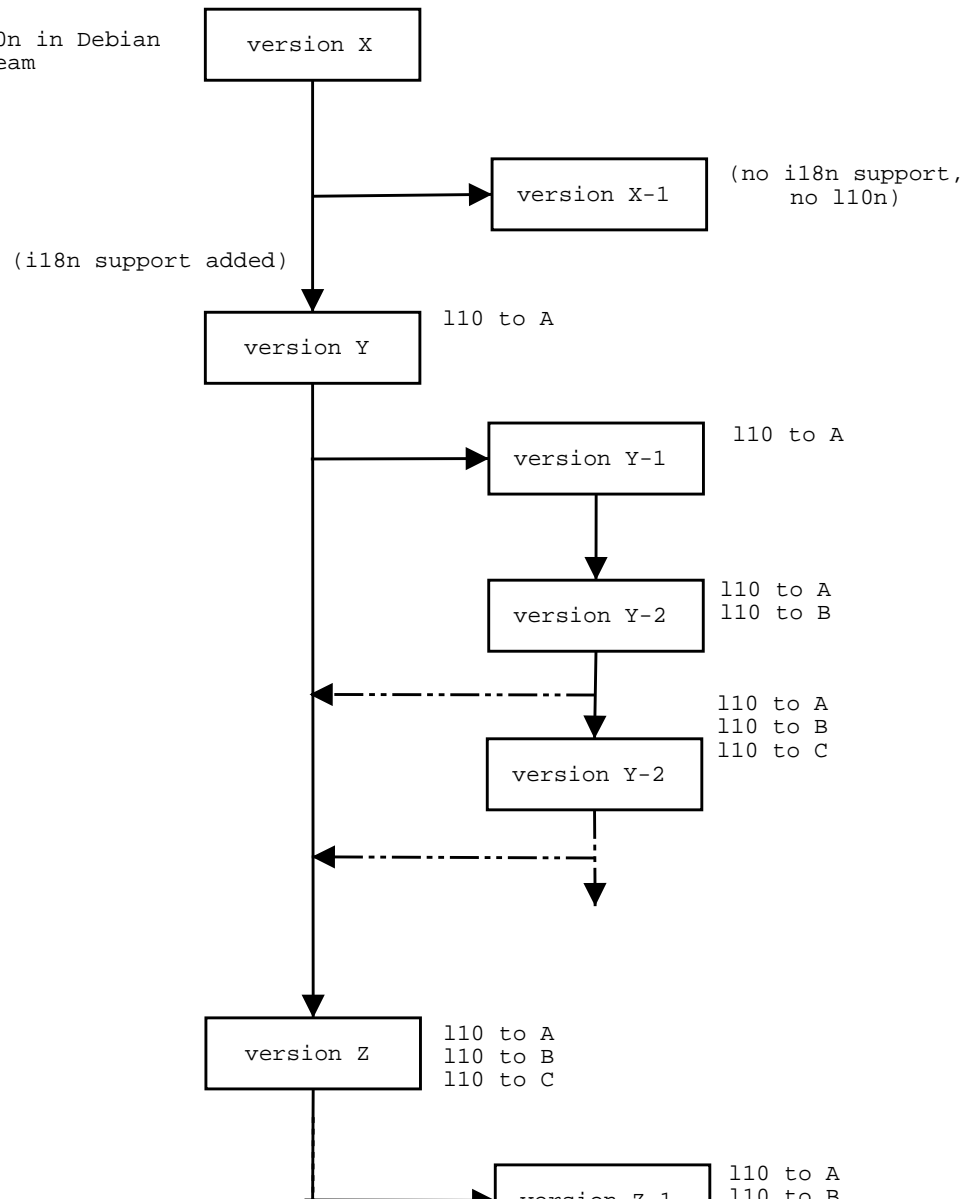# *Translation of upstream:* leave it up to them

Option 1: Leave translation
up to upstream

```
version X
```

```
version X-1
```

```
version Y
```
l10 to A

```
version Y-1
```
l10 to A

```
version Z
```
l10 to A
l10 to B

```
version Z-1
```
l10 to A
l10 to B

Translation of upstream: let's do it ourselves

Option 2: Work in l10n in Debian
and provide to upstream

version X

version X-1 (no i18n support,
                 no l10n)

(i18n support added)

version Y l10 to A

version Y-1 l10 to A

version Y-2 l10 to A
            l10 to B

version Y-2 l10 to A
            l10 to B
            l10 to C

version Z l10 to A
          l10 to B
          l10 to C

version Z-1 l10 to A
            l10 to B

Some proposals

Many questions, answers needed (discussion in the mailing lists, please)

- Focus i18n/l10n teams
- Officially supported languages
- Translation *freeze*
- Translators' status
- More tools for the i18n/l10n people
- Support of i18n/l10n tags in the BTS
- Single configuration for language

Will not detail all in this talk....

Proposal: Define priorities to focus i18n/l10n teams

- Debian as a project should say which are the *officially supported* languages:
  - Not all languages are equally well supported
  - It is needed to make decisions (sample: d-i or doc in CDs)
  - If we don't we have to use world data (not relevant)
- Prioritise work, what should i18n/l10n teams first focus on? Suggested:
  - First: installation
  - Second:documentation (release notes, manuals)
  - Third: debian-specific programs
  - Fourth: other data (such as descriptions or other docs)
  - Fifth: non-debian programs

Proposal: Translation *freeze*

- After the feature freeze introduce a new freeze for translators

- NMU for l10n related issues are accepted (and encouraged)

- Scheduled before

- Reasoning: Translation is done against a *moving target*:

  - sometimes translators cannot work as fast as upstream.

  - sometimes it does not make sense to translated until it's *finished*

- Some other projects do this (i.e. GNOME, KDE)

testing -> freeze -> translation freeze -> deep freeze -> stable

Proposal: Translators as DD

- Reasoning: translators need access to internal resources to help with translations. Otherwise they need to use proxies (slows down process)

- Make them DD with limited access?
  - No upload of packages (no NMU)
  - Access to CVS servers
  - Access to Alioth

Proposal: BTS support *for i18n/l10n*

- Add a *i18n* tag

- Add a *l10n* tag (with language subtag), see #114221

- Reasoning:
    - What should a DD do when he receives a translation-related bug?
    - Coordination of i18n/l10n integration
    - Bugs cannot be assigned to translation teams (and returned)
    - Users can send bug reports in their language and, through this tag, have it translated for the DD.

Proposal: More tools for translation teams

- Clearing house for translations
  - Single clearing-house for translations: simplify i18n work. (look #105337)
  - GNU's project is not good enough (not all apps in Debian are GNU)
- Provide consistent tools for translations teams, Debian-specific

Proposal: Simple configuration for the system langua

- Easy setup of language environment for the end user
  - Just do it once! (on installation)
  - Base it on language / timezone?
  - Use it everywhere!
- user-env packages are hacks!
- Used in:
  - /etc/environment. Or is /etc/language better?
  - Use it for system services? (ftp, mailing list static content? #43257)

Summary

- i18n and l10n is a very important topic if we want to make a Universal OS

- Translators and developers focused in i18n need appropiate tools to do their job.

- We have to discuss and approve an specific policy or set of recommendations for i18n/l10n in Debian (see Martin Quinson's proposal)