

Localization of softwares in Indian Languages

Why?

- Unavailability of computing resources and internet to majority of peoples.
- Use of softwares in government agencies including railways, bank.
- Dependency of software companies on US market.

How?

- Port existing application in Indian language.
- Translate user interface.

Internationalization

- Taking the necessary steps to make application aware of different languages and national standards.
- Operation by which a program, or a set of programs turned into a package, is made aware and able to support multiple languages.
- Generalization.

Localization

- Enabling an internationalized application to behave correctly with a certain language.
- Using generic methods already implemented in an internationalized program in specific ways.
- Particularisation process.

Internationalizing 'hello world'

- Extract strings using *xgettext*.
- Update portable object file.
- Creating machine object file.
- Using *gettext* for output.

xgettext

- Scans all the source files given as arguments, for message strings.
- Creates an editable .po file which contains all these message-strings

e.g.

```
$ xgettext -a --C --force <c-code file name>
```

Portable object files

- po stands for Portable Object.
- Consists of entries holding the relation between an original string and its corresponding translation.
- The original string is introduced by the keyword msgid, and the translation, by msgstr.
- po files can be updated using tupdate.

tupdate new.po old.po > latest.po

Machine object file

- Compile .po file to an executable form using msgfmt.

```
$ msgfmt --output-file=<filename> <file.po>
```

- Creates the .mo files.
- export \$LANG as hi_IN.
- Place .mo file in the directory structure /usr/share/locale/hi/LC_MESSAGES.

Internationalizing program

- include header file `<libintl.h>`
- Add `gettext` for every `printf`.
eg. We need to modify

```
printf("This prints the translated text");
```

to

```
printf(gettext("This prints the translated text"));
```

Source code

```
#include<stdio.h>  
#include<locale.h>  
#include<libintl.h>  
#include<langinfo.h>  
  
main()  
{  
  setlocale(LC_ALL, "");  
  textdomain("messages");  
  printf(gettext("Welcome\n"));  
}
```

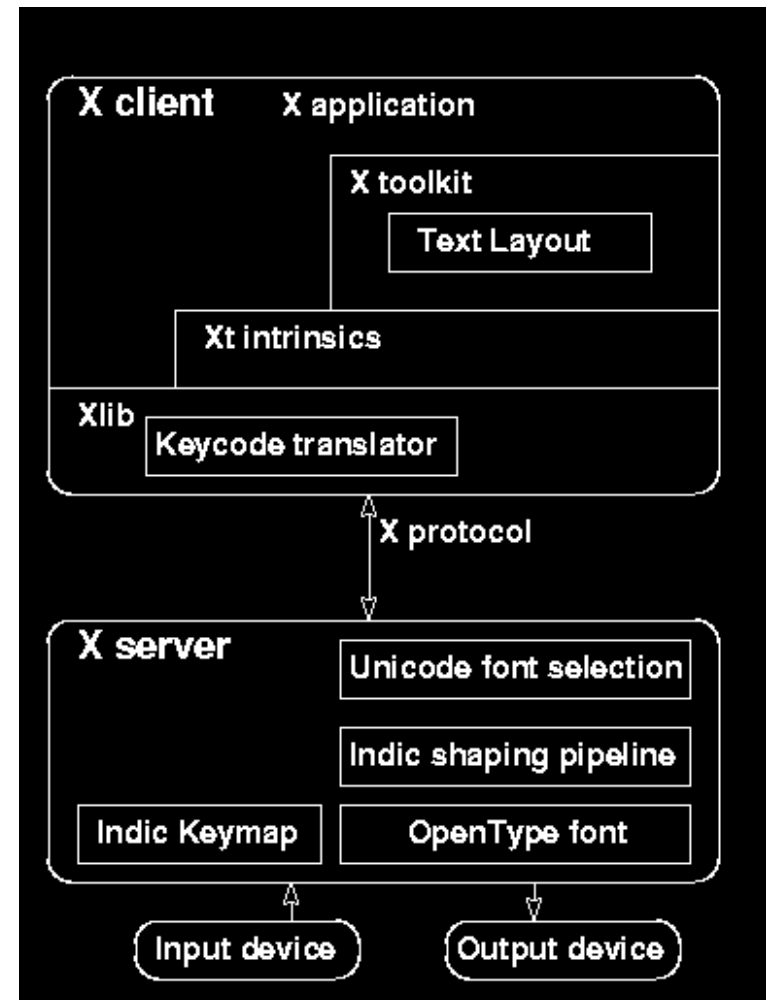
po file

```
# SOME DESCRIPTIVE TITLE.
# Copyright (C) YEAR THE PACKAGE'S COPYRIGHT HOLDER
# This file is distributed under the same license as the PACKAGE package.
# FIRST AUTHOR <EMAIL@ADDRESS>, YEAR.
#
#, fuzzy
msgid ""
msgstr ""
"Project-Id-Version: PACKAGE VERSION\n"
"Report-Msgid-Bugs-To: \n"
"POT-Creation-Date: 2004-04-18 05:07+0000\n"
"PO-Revision-Date: YEAR-MO-DA HO:MI+ZONE\n"
"Last-Translator: FULL NAME <EMAIL@ADDRESS>\n"
"Language-Team: LANGUAGE <LL@li.org>\n"
"MIME-Version: 1.0\n"
"Content-Type: text/plain; charset=UTF-8\n"
"Content-Transfer-Encoding: 8bit\n"

#: test.c:10
msgid "Welcome\n"
msgstr "स्व।गतम्"
```

Efforts for localization: Indic

- Basic changes in X Windows system infrastructure.
- Ease in localization of individual packages.
- Treat 8 bit string as UTF -8 encoding of unicode characters.
- No effect in representing english text.

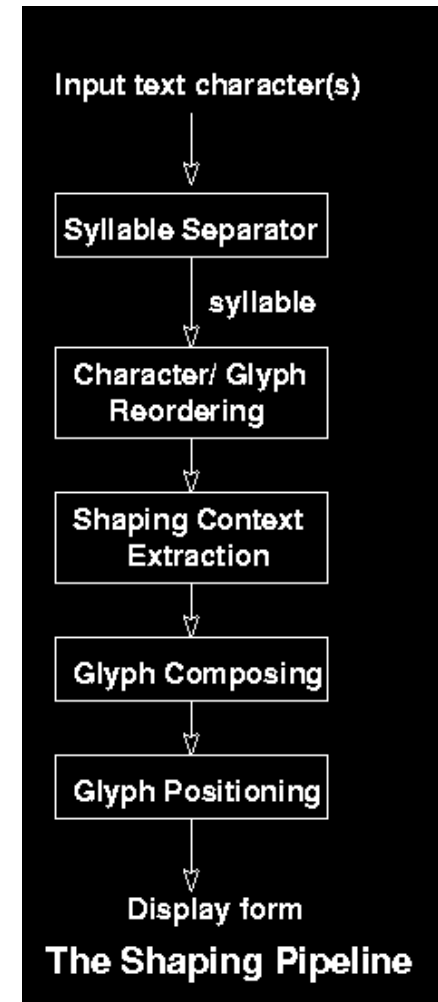


OpenType font support : Indix

- Phonetic nature of the Indian scripts, results in change of character position and appearance.
For example, क् र should be displayed as क्र and not as क् र.
- Use of OpenType font for substitution and positioning information required in Indian scripts.

For example,

- Text sequence क् र म is typed by the user.
- Syllables क् र and म.
- Reordering first syllable as क र्.
- Substitution gives क्र.
- Positioning these two glyphs as क्रम.



What can I do?

- System level programming:
e.g.
 - Support of unicode in display/printing
 - Creating fonts with embedded rules(opentype)
 - Creating the fonts that will be acceptable to all.
 - Testing i.e quality assurance.

What can I do?

- Application level programming:
 - Porting of software to new libraries that support unicode and indian languages.
 - Preparing manuals, tutorials for applications.
 - Translating the text strings and documentation.

What can I do?

- Content level:
 - Translation of manual, tutorials and articles.