Going international

using Unicode's CLDR Project to localise your software

Petrus Janse van Rensburg - ElixirConf Africa, 7 October 2022

i18n & l10n

i18n - Internationalization

The process of designing a software application so that it can be adapted to various languages and regions without engineering changes.

L10n - Localization

The process of adapting internationalized software for a specific region or language by translating text and adding locale-specific components.

But what is a locale?

Locale: Language code + Country code

E.g. `"en-US", "en-UK", "en-ZA"` etc.

```
"en" == "en-US"

"fr" == "fr-FR"
```

But there's more:

- Number format setting
- Date-time format setting
- String collation setting
- Currency format setting

```
"en-AU-u-tz-ausyd-cf-account"
```

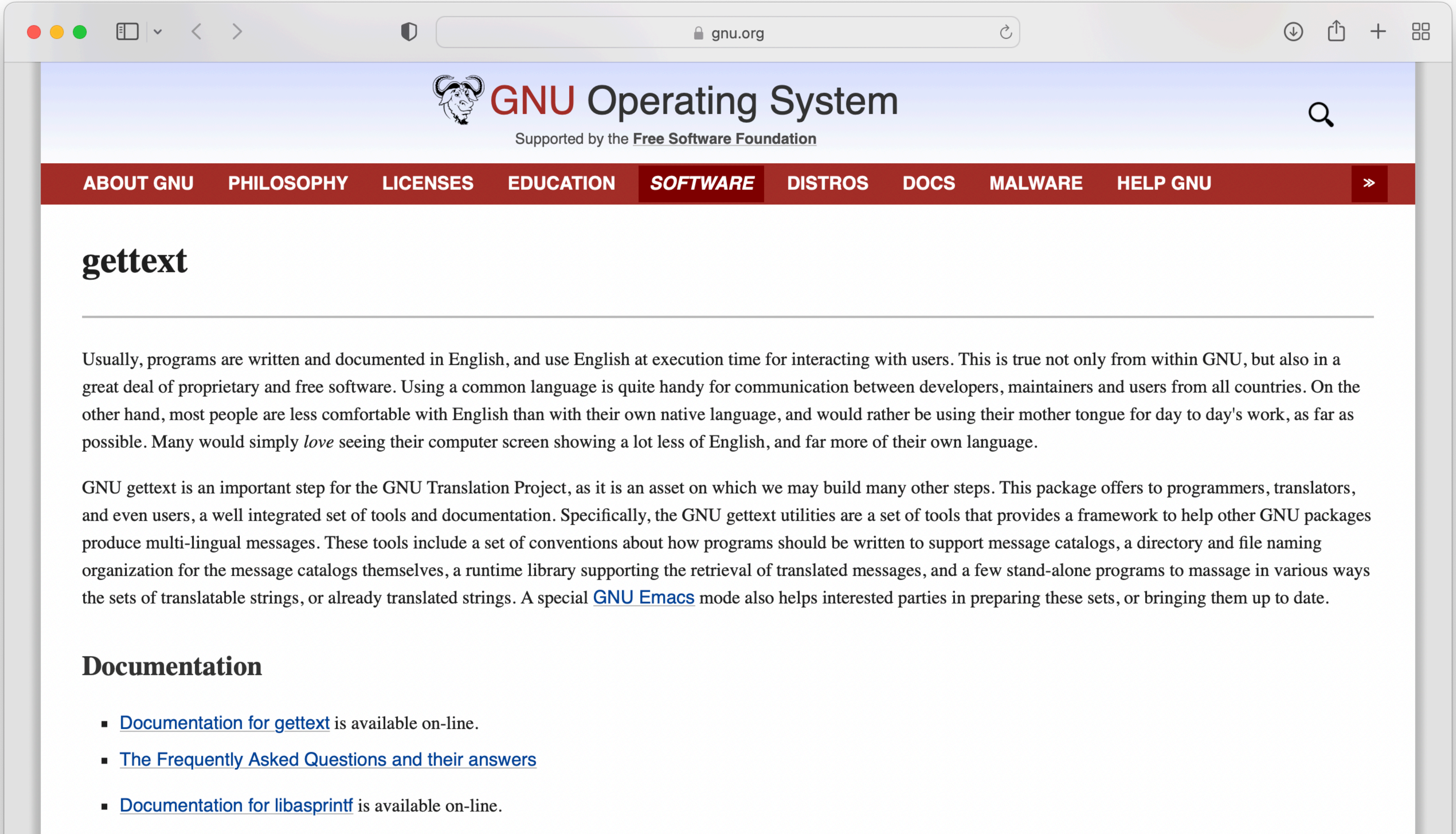
- language: English
- territory: AU
- timezone: "Australia/Sydney"
- currency format: accounting

Your software is running on some locale or another.

The Unicode Consortium

- https://en.wikipedia.org/wiki/Unicode_Consortium
- https://home.unicode.org/
- https://cldr.unicode.org

Step 1: gettext



gettext

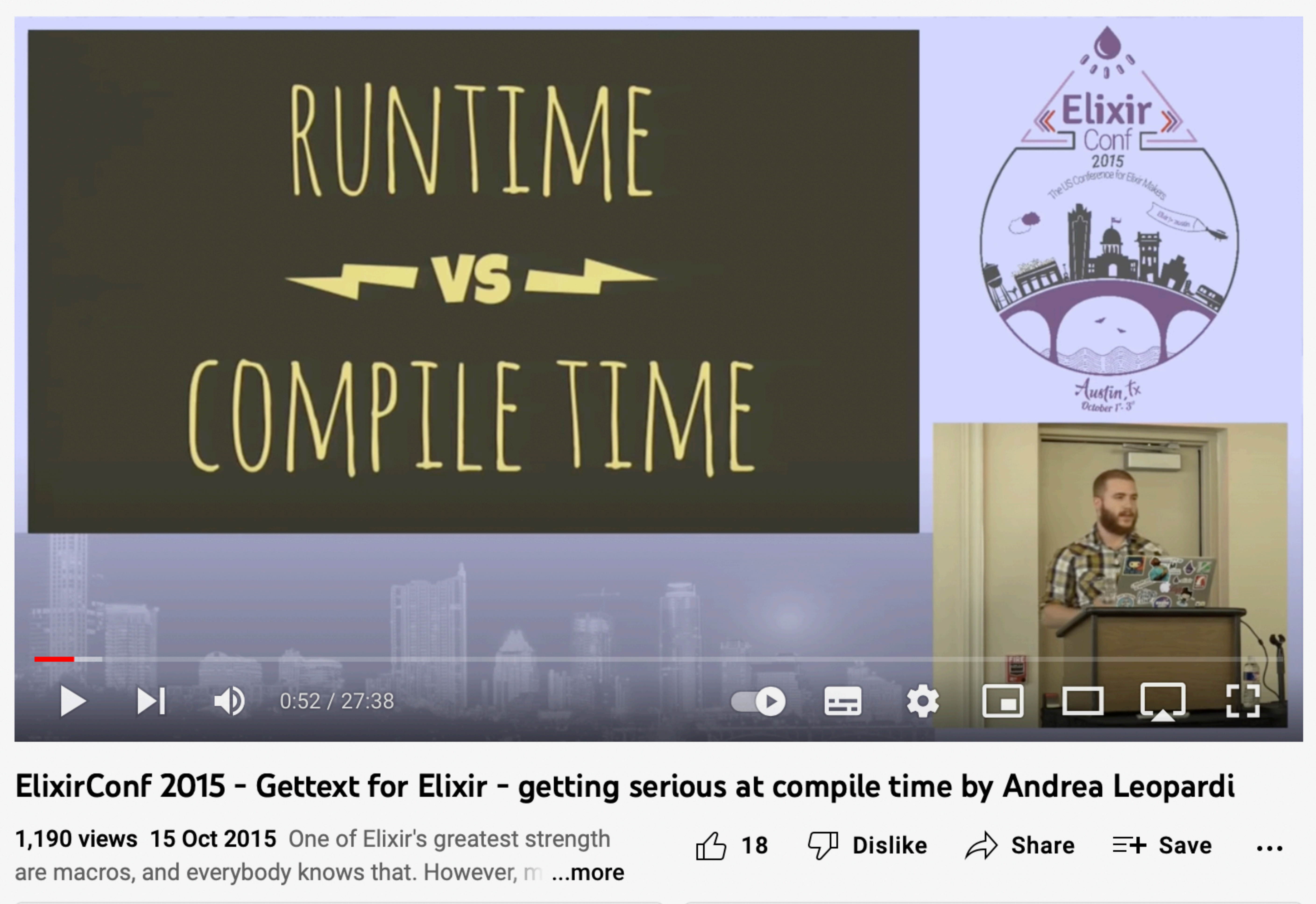
Usually, programs are written and documented in English, and use English at execution time for interacting with users. This is true not only from within GNU, but also in a great deal of proprietary and free software. Using a common language is quite handy for communication between developers, maintainers and users from all countries. On the other hand, most people are less comfortable with English than with their own native language, and would rather be using their mother tongue for day to day's work, as far as possible. Many would simply *love* seeing their computer screen showing a lot less of English, and far more of their own language.

GNU gettext is an important step for the GNU Translation Project, as it is an asset on which we may build many other steps. This package offers to programmers, translators, and even users, a well integrated set of tools and documentation. Specifically, the GNU gettext utilities are a set of tools that provides a framework to help other GNU packages produce multi-lingual messages. These tools include a set of conventions about how programs should be written to support message catalogs, a directory and file naming organization for the message catalogs themselves, a runtime library supporting the retrieval of translated messages, and a few stand-alone programs to massage in various ways the sets of translatable strings, or already translated strings. A special GNU Emacs mode also helps interested parties in preparing these sets, or bringing them up to date.

Documentation

- Documentation for gettext is available on-line.
- The Frequently Asked Questions and their answers
- Documentation for libasprintf is available on-line.

- https://en.wikipedia.org/wiki/Gettext
- https://poeditor.com
- https://github.com/elixir-gettext/gettext



ElixirConf 2015 - Gettext for Elixir - getting serious at compile time by Andrea Leopardi

1,190 views 15 Oct 2015 One of Elixir's greatest strength are macros, and everybody knows that. However, m ...more

https://www.youtube.com/watch?v=liP_cTfzk_o

gettext workflow

markup > extract > translate > compile

but, gettext has limitations

Step 2: CLDR

Dates

Numbers

Currencies

DEMO

- https://github.com/unicode-org/cldr-json
- https://github.com/elixir-cldr
- https://fco.e23delivery.com/splash

Thank You