

Интернационализация (i18n) простой текстовой разметки *Подход с использованием **gettext***

Поташников Николай

Version 1.0, 2022-01-11

Содержание

Введение	4
Основная идея	5
Описание процесса	6
Исходные шаги	6
Обновление перевода	7
Несколько замечаний	8
Заключение	9



Введение

[Go to English version](#)

Несколько лет назад коллега опубликовал [статью](#) о создании презентации при помощи [Asciidoctor](#). С тех пор мы пользуемся только этим подходом.

Недавно возникла проблема перевести презентации на несколько языков и обеспечить синхронизацию переводов по мере доработки оригинала. Решение оказалось настолько простым и отработанным, что я решил описать его в этой статье.

Предлагаемое решение ничего не знает о синтаксисе. Это означает, что не так важно, используется AsciiDoc, другой формат простой текстовой разметки или даже смешанный формат (например, включающий мои любимые Plantuml или любые другие диаграммы). Серьезные ограничения данного подхода — (1) переводчик не должен ломать используемую разметку и (2) невозможно напрямую использовать машинный перевод.

Решение использует [Translation Toolkit](#) и стандартные [инструменты GNU Gettext](#).

Для демонстрации подхода у данной статьи есть английская и русская версия. В её [репозитории](#) настроена простая автоматизация, которая синхронизирует перевод, создает печатную версию статьи (pdf, docx, odt), а также создает файл в формате Markdown для публикации на Хабре.

В [предыдущей статье о тестировании документации](#) я только упомянул стандартные текстовые линтеры, т.к. старался сфокусироваться в целом на подходах к тестированию, а не конкретных инструментах. Тем не менее существующие линтеры могут очень и очень многое. Поэтому эта статья проверялась при помощи [vale](#).

Основная идея

Gettext предполагает, что ключевые строки для перевода являются оригинальными сообщениями.

Gettext использует файлы расширением `.po` (PO — [Portable Object](#)) для хранения как оригинальных, так и переведенных сообщений. Большое количество редакторов позволяет редактировать такие файлы как в однопользовательской, так и в многопользовательской среде.

Основная идея Translation Toolkit заключается в использовании блоков смежных строк в качестве констант для перевода.

Рассмотрим пример:

```
.Зима -- это

* снег
* мороз

* Рождество
* Новый год
```

В примере показано три блока смежных строк, поэтому Translation Toolkit извлечёт три константы для перевода и поместит их в файл с расширением `.pot` (шаблон `.po`)

Вставляя или убирая переносы строк в данном примере вы можете получить любое количество констант в диапазоне от 1 до 5. Количество констант зависит от удобства для переводчика.

Используя файл `.pot` в качестве шаблона, Gettext создает (обновляет) файлы `.po` для всех требуемых языков. Переводчики работают именно с этими файлами. Далее Translation Toolkit берет (1) файл `.po` с переводом, (2) оригинальный файл и создает переведенный файл.

Описание процесса

Процесс перевода состоит из следующих шагов.

- Исходные шаги для получения первого перевода на один или несколько языков.
- Обновление перевода после модификации исходного текста.

На следующих диаграммах предполагается, что исходный файл — `i18n-adoc.adoc`, а перевод должен быть помещен в файл `i18n-adoc-ru.adoc`.

Исходные шаги

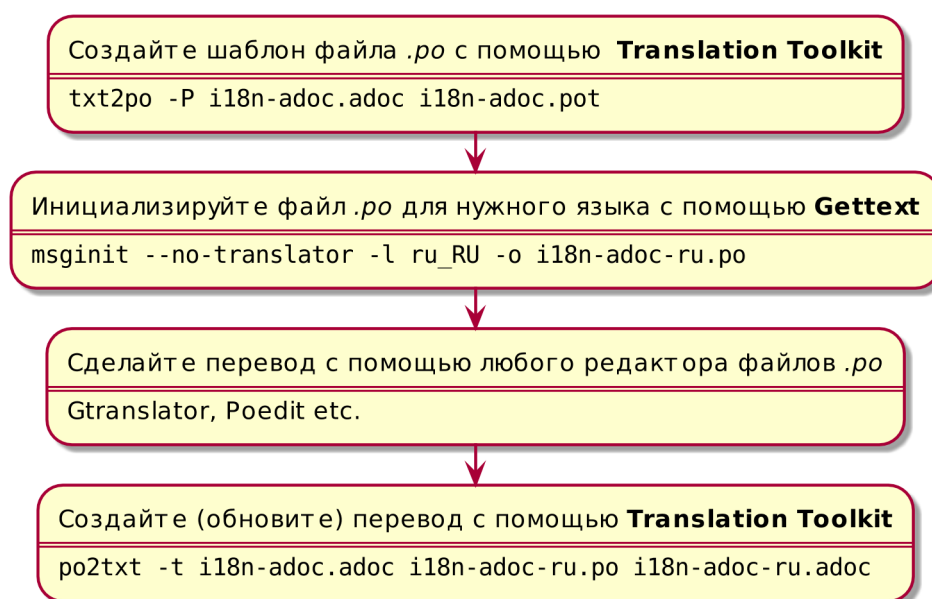


Рисунок 1. Исходные шаги

Существует много редакторов для перевода файлов `.po`. На следующем снимке экрана показан интерфейс [Gtranslator](#). Я предпочитаю [Poedit](#), хотя то, как он заменяет гравис на апостроф при использовании машинного перевода, раздражает.

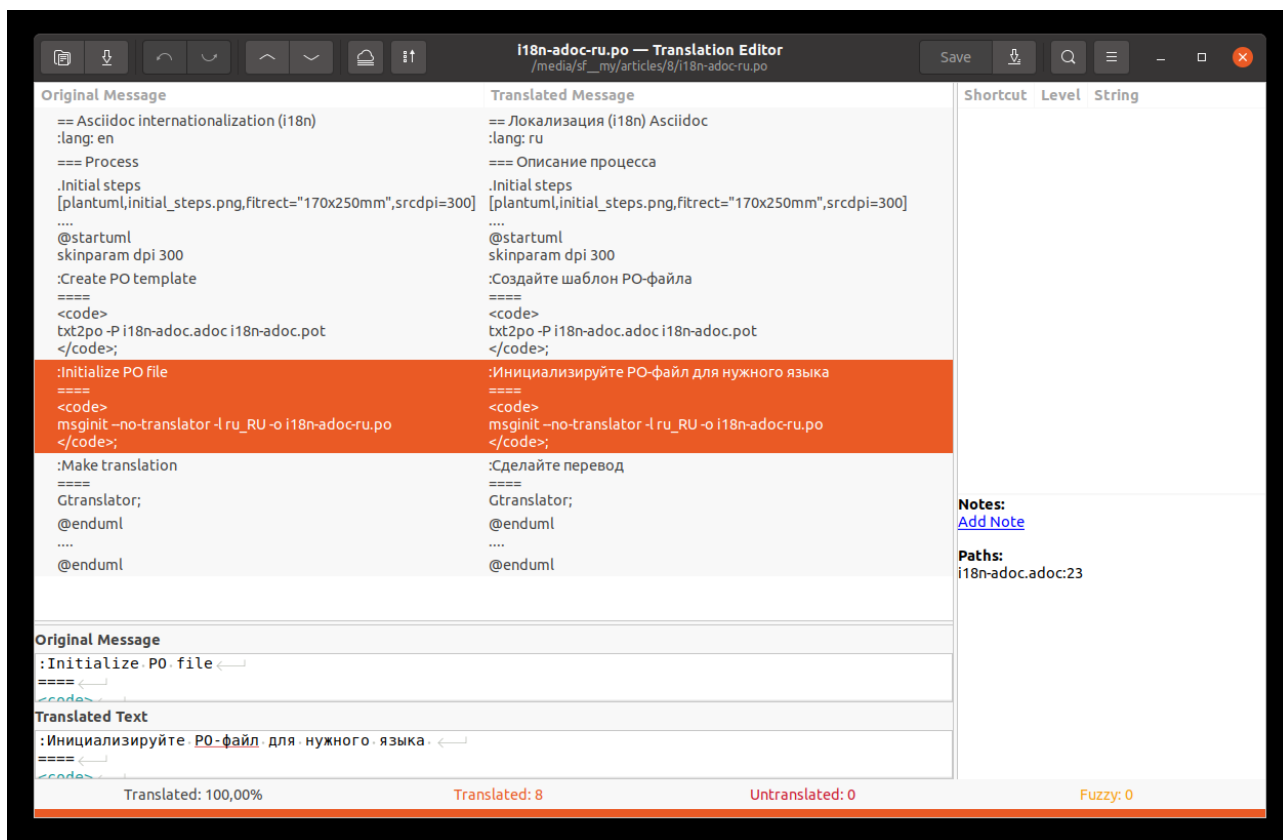


Рисунок 2. Перевод с использованием Gtranslator

Обновление перевода

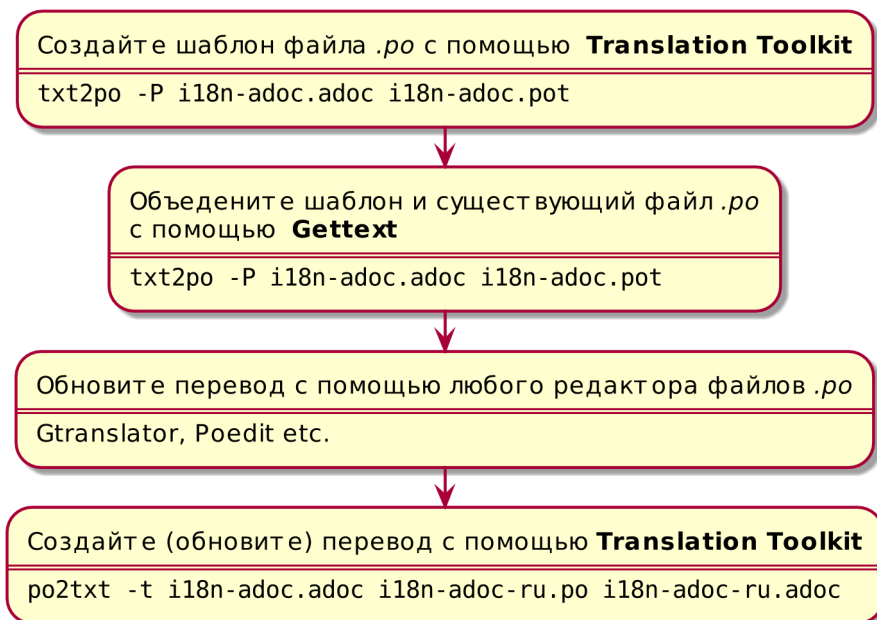


Рисунок 3. Обновление перевода

Несколько замечаний

1. В нашей документации мы часто повторно используем константы интернационализации, чтобы имена элементов интерфейса в документации совпадали с именами этих же элементов в приложении. Мы генерируем эти константы автоматически в следующем формате:

```
:main-menu-documents: Документы
:main-menu-documents-my: Мои
...
```

Мы включаем этот файл в документ Asciidoc (`include i17n-{lang}.adoc[]`). Теперь нет необходимости использовать атрибуты. Просто переведите `include i17n-en.adoc[]` в `include i17n-ru.adoc[]`.

2. Когда `gettext` обновляет файлы `.po`, он использует нечеткий поиск. Если вы немного измените исходный текст, перевод не пропадет. Он будет просто помечен как возможно неверный (flaky).
3. Актуальность файла с переводом проверить очень легко при помощи утилиты `Gettext` — `msgfmt`.

```
msgfmt --statistics i18n-adoc-ru.po
```

Команда показывает количество переведенных строк, количество строк, которые нуждаются в проверке, и количество непереведенных строк.

Заключение

- Translation Toolkit и Gettext обеспечивают эффективную интернационализацию документации.
- Простая текстовая разметка — не такая уж и простая. Использование всех возможностей простой текстовой разметки выставляет определенный уровень требований к специалисту по документации. Попробуйте представить переводчику файлы в формате .po. Многие ли из них будут готовы сделать перевод? Или попросят предоставить текст в более традиционном формате, например, в Microsoft Word.
- Контроль качества: 58 переведенных сообщений, 0 ошибок, 0 предупреждений и 0 предложений в 1 файле.