

Best practices guide for international markets ready products

TABLE OF CONTENTS

Introduction	4
Globalization, Internationalization, Localization & Translation	5
Globalization (G11N)	5
Internationalization (I18N)	5
Localization (L10N).....	5
Translation (T9N).....	5
The best practices and common issues.....	6
Poor quality source text = poor quality translation.....	6
Clarity and Brevity.....	6
Acronyms and Abbreviations	6
Proofreading and Spellchecking	7
Units of Measurement, Currency Symbols, Other Location-dependent Standards.....	7
Context is the key!.....	7
The THOR principle.....	8
Terminology Management	9
Along the development process... ..	10
Pseudo-translation.....	10
Character Encoding.....	10
Unique String ID.....	10
Placeholders.....	10

INTRODUCTION

The purpose of this document **IS NOT** to give a step-by-step guideline to get a translation done quickly.

The purpose of this document **IS** to help you develop products that are translation-friendly. Even though developing in a translation-friendly way might demand a little shift of mindset and some initial effort, you will have sustainable advantages in the long run:

- The translation process will only be the final step - and for each new release, follow the predictable cycle: source file extraction, translation process, translation import.
- The expected translation turnaround time can be reliably estimated.
- Translation bug reporting and fixing will be at minimum level.
- The translation source files stay relatively stable and the translation memory can be leveraged as much as possible, saving cost and time.
- If you decide to enter a new market with a previously not supported language, it will be easy and safe to get the whole product translated into the new language.

GLOBALIZATION, INTERNATIONALIZATION, LOCALIZATION & TRANSLATION

Globalization (G11N)

Globalization consists of internationalization, translation and localization. In order to take your product to global market and reach wider audience across globe, it is important that you also provide your product into different locales. Many product based companies have been very actively globalizing their product. Product in regional language attracts more number of customer, Globalization makes it easier for people to use product in their regional language.

Internationalization (I18N)

Internationalization is the process of designing a software product with respect to future adaptation to target languages and regions. It is important to see this process as a mean to avoid development/code changes effort in the future.

Localization (L10N)

Localization is the process of adapting internationalized software for a specific region or language by translating text and addressing cultural and non-textual components, as well as linguistic issues.

Translation (T9N)

Translation is the process of converting text into another language.

THE BEST PRACTICES AND COMMON ISSUES

Poor quality source text = poor quality translation

Developers are the masters of their universe. When it comes to writing the strings, they should not be tasked with making sure the source string text is of a good quality and well translatable.

Recommendation:

Have a technical writer to perform a sign-off of the text. Such a person should be trained on what translators need to carry out the translation.

The biggest benefit will be a good quality, consistent source, as well as lower number of linguistic queries during translation and shorter translation turn-around-time.

Clarity and Brevity

- Complex sentences are always a pitfall for translation. It's always better to write three short sentences instead of one long sentence.
- While brevity is advised, keep in mind that translators need to understand the meaning and relations of the words:
 - write complete sentences (e.g. instead of "Display selected" write "Display selected records")
 - provide context notes for short strings (e.g. "open bill" write is this a "bill" which is "open", or a button in order to "open" a "bill"?)
 - avoid noun chains (try to avoid noun chains such as "customer case request time limitation")
- Don't use humor or irony.

Acronyms and Abbreviations

- Try to avoid acronyms and abbreviations:
 - Don't use short forms of your terminology - just because it makes sense in English, it might still cause issues, mis references and inconsistencies in the translations.

Proofreading and Spellchecking

- Use a spellchecker.
- Let someone else read your UI-strings.
- It's advisable to let a person with language experience (copywriting or translating) read your strings.

Units of Measurement, Currency Symbols, Other Location-dependent Standards

- Avoid of hard-writing any of these in a UI-string. If not handled via placeholders it can be a great risk to have e.g. a Dollar symbol in a UI string - even if the user switches the currency of the transaction, such a UI-string would cause confusion.
- If you for some reason definitely need to hard-write such a symbol/standard into your UI-string, indicate in the context note something like this: "This has to stay a Dollar symbol, no matter what" - or - "This can be adjusted to the locally predominant currency symbol".

Context is the key!

The translators usually do not get to see the product itself. And even if they do, they are not as familiar with the product, as the authors are. Translators will have difficulties understanding where the given string is rendered in the UI and what context should be taken into the account during translation. Therefore, it is wise, to have a mechanism of accompanying a string with its contextual information. Technically, it can be "note" attribute in XML/XLIFF, it can be specifically formatted text in rich text format, etc.

Also keep in mind, the CAT (Computer Aided Translation) tool will usually display only the text that is to be translated. All the tagging and coding around is for the machines to read, not the human.

Example:

Translator needs to translate one word string “State”. In the best case, he/she will send a linguistic query – asking for the context. In the worst case, the translator will translate such a string with the best guess.

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xliff xmlns="urn:oasis:names:tc:xliff:document:1.2" version="1.2">
  <file datatype="plaintext" source-language="en-US" original="xliff-1.2-strict.xsd">
    <body>
      <trans-unit id="State_geo" translate="yes">
        <source>State</source>
        <note>This is a geographical location.</note>
      </trans-unit>
      <trans-unit id="State_switch" translate="yes">
        <source>State</source>
        <note>This is a state of a switch.</note>
      </trans-unit>
    </body>
  </file>
</xliff>

```

Figure 1: Example of well written contextual information

The THOR principle

For the creation of a context note, you can follow the THOR principle:

T	Terminology (domain)	A word can have different meanings and translations in different domains. If you can identify a UI-string as belonging to "Inventory Management" or "Project Management", etc., the translators already know which translation to choose.
H	Heart (part) of speech	A UI-string itself sometimes cannot indicate whether it is a verb or a noun (for example, Store). This is also the right place to explain the meaning of an acronym/ abbreviation and whether or not it should be a DNT (Do-Not-Translate)
O	Object (in the UI)	The translation can differ depending on the role the UI-string has in the bigger picture in the UI, for example, if the string is the label of a button or the title of a page, e.g. Search .
R	Reference (what happens with it?)	Here, the reference of the UI-string can be explained. If it is a title of a page, what is displayed on the page? If it is a button, what happens when clicking the button? Especially important is if the UI-string contains placeholders such as { 1 } - what kind of values will be inserted there?

Terminology Management

What kind of Terminology should be managed at minimum?

- Key terms of the all products and departments
 - Product names
 - Branded phrases/slogans
- General terms/industry terms with a special meaning in your product
- Abbreviations/short forms for terms
- Acronyms
- DNT (Do Not Translate Terms) - the name of your own technology, coding language, connectors, etc.

ALONG THE DEVELOPMENT PROCESS...

There is number of activities developers should carry out during the development to ensure ease of translation.

Pseudo-translation

By applying pseudo-translation, it is possible to spot issues and fix them, before the strings are sent for translation.

It is a process of replacing vowels with their accented variants, doubling some characters to make the string longer, putting custom characters at the beginning and at the end of the string, so either a human or an automation tool (like Selenium) can spot the issues.

^\$^Thís mááyý béé áán ééxáámpléé óóf psééúúódóó tráánsláátééd sééntééncéé.^#^

When a human or a machine detects an unaltered text after pseudo-translation has been applied, it is probably a hard-coded text, that failed to get extracted for translation and a code change needs to happen.

Character Encoding

Make sure your product is ready to support all possible characters. Preferably, follow the Unicode standard.

Unique String ID

By implementing a “unique string ID” concept, you will gain following benefits:

- When raising a linguistic query, translators can refer to an ID instead of the content of the string.
- Automation can be built to perform testing of your product, crawling through the product screens, performing user actions, while capturing screenshots for every screen, storing the screenshots along with strings found into a repository. Such a repository can then be made available to translators, who can look up all screens where given string is used.

Placeholders

If the string contains a placeholder, the translators need to know which words/values can be inserted via this placeholder. This information can also be included in the placeholder.



L10N Best Practices Guide 2020