Localization Processes and Best Practices for Web Developers and Translators

With an Evaluation of Localize, a Translation Project Management Application

Kimberly Hawthorne

FTRA-636

Informatique et traduction

Concordia University

Département d'études françaises

April 14, 2020

Though English has long been the dominant language on the Web, the ratio of English-speaking users has been steadily declining since 2000 as the world becomes more and more connected (Internet World Stats 2020). Such diversification of users has led to an increase in the number of languages on the Web, and with that, an increased need for localization. While its definition remains debated in the field of translation studies, localization is generally accepted to be "the process of adapting a product or content to a specific locale or market," which includes but is not limited to translation of textual content (Globalization and Localization Association, n.d.). Many different products—including software, video games and multimedia content—may require localization, but this research will focus on websites (Jiménez-Crespo 2013, 28). Localization processes and best practices will be examined from the perspective of Web developers and translators, and with these considerations in mind, an online localization management tool called Localize[1] will be evaluated.

The process of localization

According to Miguel Jiménez-Crespo (2013, 29-31) in his study of Web localization, the localization process can be broken down into three main stages: project preparation, translation and quality assurance (QA). A successful Web localization project requires careful planning and involves professionals from a multitude of disciplines, including project managers, copywriters, translators, programmers, user experience (UX) designers and QA specialists (Jiménez-Crespo 2013, 29). While localization mainly consists of textual language transfer, several other factors must be considered at every stage of the process. The first two—project preparation and translation—are discussed below.

[1] https://localizejs.com/

Language transfer and cultural adaptation

Perhaps most obviously, text present in website interfaces needs to be translated in a localization project. Both copywriters and programmers should be cognizant of differing grammatical structures among languages when preparing text segments for localization. Beyond language transfer, a website may also require cultural adaptations. If the goal of localization is "to produce products that are received by target users as locally made," elements like date formats, units of measurement and currencies should be adjusted to fit the target locale (Jiménez-Crespo 2013, 14; MDN Web Docs 2020).

In preparing text for localization, programmers should strive for neutrality, making no assumptions about the target language. The need for neutrality is apparent when considering differences in pluralization and gendered nouns among languages—especially when dealing with dynamic data. According to the Mozilla Developer Network documentation (2020), "if a string will contain a placeholder [for dynamic text], always add the placeholder to the string to allow the localizer to change the word order if necessary." At first glance, developers may be tempted to extract placeholders from translation segments and concatenate translated strings, in an effort to simplify the codebase and save time in translation. However, such an approach can become unwieldy or even impossible when supporting multiple languages. Take the following example, in which the country changes based on locale:

```
shipping-information-label: {
   en-CA: "Shipping information for ${country-adjective} addresses",
   fr-CA: "Informations sur la livraison pour les adresses
${country-adjective}",
   fr-FR: "Informations sur la livraison pour les adresses
${country-adjective}"
},
country-adjective: {
   en-CA: "Canadian",
   fr-CA: "canadiennes", // ERROR: assumes gender and pluralization
   fr-FR: "françaises" // ERROR: assumes gender and pluralization
}
```

With this setup, the English source segment ("Shipping information for ${country-adjective} addresses") would only need to be translated into French once, as the adjectives *canadiennes* and *françaises* could be programmatically updated to match the selected locale. In French, the country adjectives would necessarily be translated and stored in the feminine and plural forms, to agree with the noun *adresses*. The impetus behind such an approach would be to reuse the "country-adjective" variable in other segments; however, since the French adjectives are "hard-coded" to be both feminine and plural, they run the risk of rendering grammatically incorrect content in other contexts. Thus, string concatenation of dynamic placeholder text makes assumptions, often incorrect, about target grammatical structures and should be avoided. Instead, translation segments should comprise complete phrases, allowing translation professionals to handle all grammar logic.

```
shipping-information-label: {
   en-CA: "Shipping information for Canadian addresses",
   fr-CA: "Informations sur la livraison pour les adresses
canadiennes",
   fr-FR: "Informations sur la livraison pour les adresses françaises"
}
```

While Jiménez-Crespo (2013, 30) indicates that translators should be provided with notes, sample folder structures and even a localization environment of the website, such resources are unfortunately not always available. As such, developers should strive to provide translators as much contextual information as possible when preparing text segments for localization.

User interfaces

Localization requires not only programmatic changes, but also UX considerations. Ideal font sizes and text expansion may differ between languages and impact designs, requiring flexible or even completely different interfaces. For example, font sizes may need to be increased for websites that support languages with complex characters, such as Korean and Japanese (W3C Internationalization 2016a). Additionally, UX designers would need to be heavily involved in localizing websites that support languages with various text directions, such as a multilingual website that supports both English (left to right) and Arabic (right to left) (W3C Internationalization. 2016b).

Accessibility

Finally, accessibility labels and elements must also be localized to ensure that all users in the target locale can access the website, regardless of ability. This is especially important when localizing websites that are accessible to visually impaired users, many of whom navigate the Web using screen-reader software. According to the W3C Web Accessibility Initiative (2017), screen readers convert visual content—including text, icons, images and the state of interactive elements—into a communicative form that the user can consume, such as speech or Braille. To ensure compatibility with screen readers, developers make use of ARIA attributes. These attributes can be added to HTML tags to provide additional context for visually impaired users via “roles and states that describe the behavior of most familiar UI widgets” (MDN Web Docs 2019).

Presumably, a website with accessible components aims to be available to all of its potential users. Thus, if that website supports multiple languages, its accessibility functionality should also be included in a comprehensive localization project. As previously mentioned, translators should be provided additional context about these accessibility segments. Since ARIA content is visually hidden, the length of the translated text will not affect the website’s layout; therefore, translators should know to prioritize clarity in these instances (MDN Web Docs 2020). Such information can be conveyed via localization notes directly in the translation files.

```
delivery-options-aria-label: {
     en-CA: "Click here for delivery options" // LOCALIZATION NOTE:
used for accessibility and is visually hidden
}
```

Evaluation of Localize JS

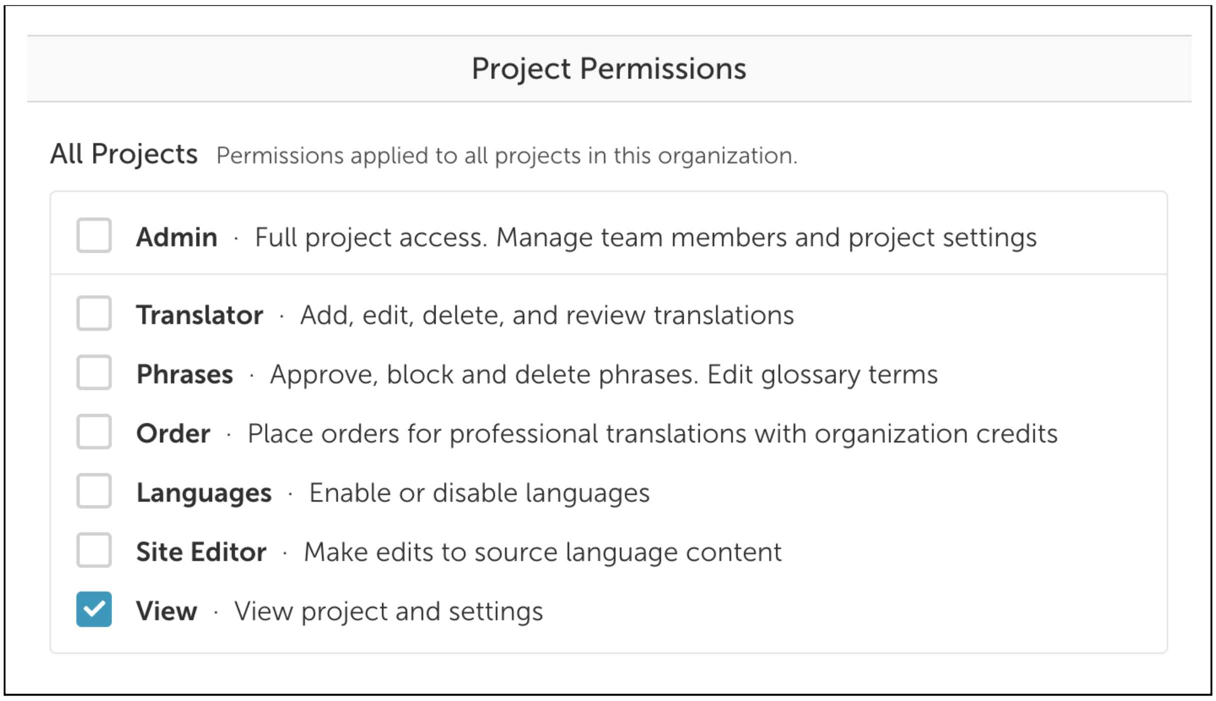
As demonstrated above, professionals involved in localization must consider several factors in addition to language transfer. With an increased demand for Web localization, many software products have been developed claiming to streamline the localization process, including Localize, which is a cloud-based translation management and delivery platform. Localize offers a wide range of features to a multitude of user types, including project managers, translators, reviewers and programmers. The application is quite rich in features, with support for machine translation, reporting, glossaries, and style guides; however, it seems to excel simply as a project management platform for localization projects.

Localize was evaluated by the author via two separate test projects. In the first project, text from the first two chapters of Antoine de Saint-Exupéry's *Le Petit Prince* was published to a test website.[2] Localize's content delivery feature was used to fetch untranslated segments directly from the website, as well as publish the approved translations. The second project followed a more traditional translation workflow in which text segments were imported into Localize using a JSON file containing test data.[3] Findings from both projects are described below.

[2] https://kimberlyhawthorne.github.io/ftra636/
[3] https://github.com/kimberlyhawthorne/ftra636/blob/master/AddToCart.json

Project management

Out of the box, Localize is an easy-to-use translation project management platform with an intuitive user interface. Administrators can easily create new translation projects, select multiple supported languages and run reports on the project's progress. They can also add users and lock down their permissions based on role. For example, translators can be limited to accessing and editing untranslated content, while revisers may be granted more freedom to not only translate, but also approve suggested translations and maintain glossaries (see Figure 1).



Project Permissions

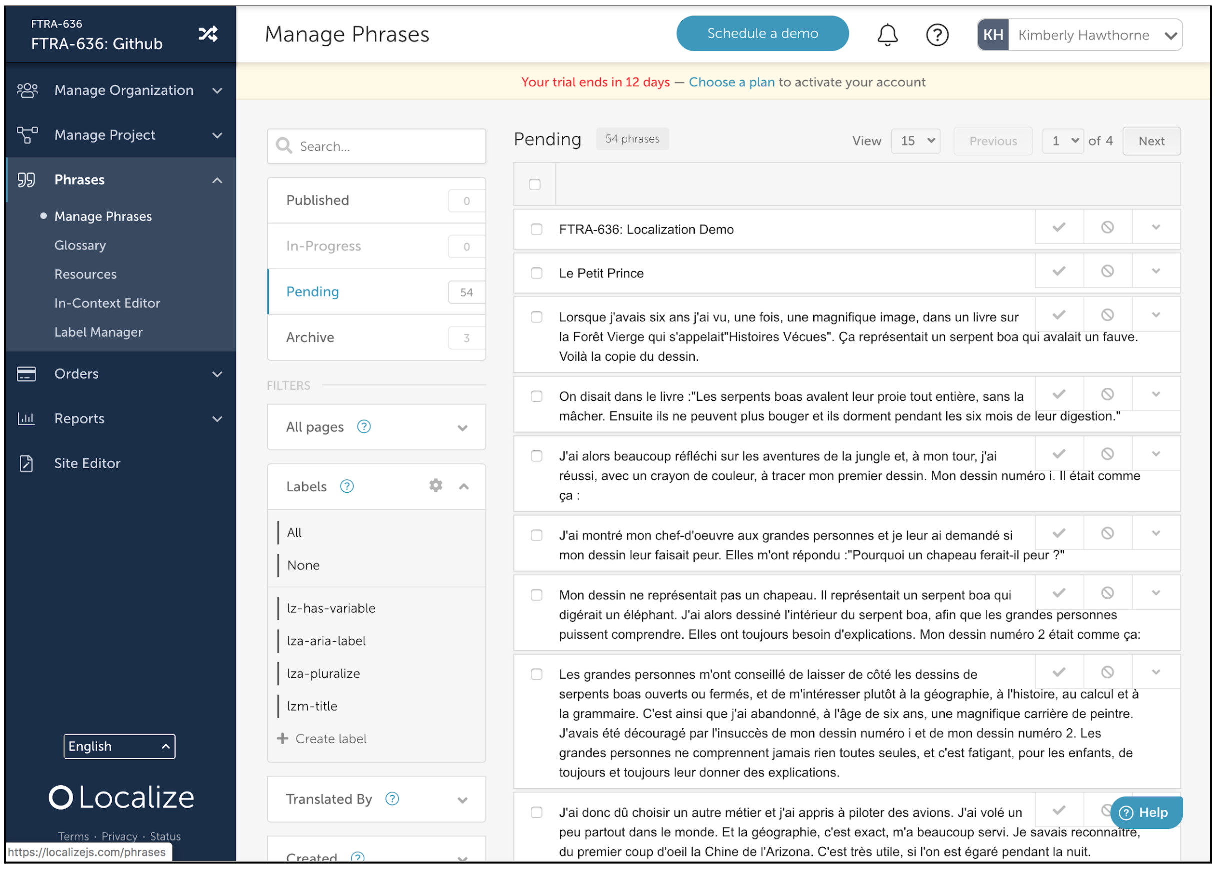
All Projects Permissions applied to all projects in this organization.

- [ ] **Admin** · Full project access. Manage team members and project settings
- [ ] **Translator** · Add, edit, delete, and review translations
- [ ] **Phrases** · Approve, block and delete phrases. Edit glossary terms
- [ ] **Order** · Place orders for professional translations with organization credits
- [ ] **Languages** · Enable or disable languages
- [ ] **Site Editor** · Make edits to source language content
- [✓] **View** · View project and settings

Figure 1: User roles and permissions in Localize.

Translation segments can be loaded into Localize in one of three ways: via a data file, an integration with a third-party tool or Localize's content delivery feature. The first option is the most traditional and, for complex websites such as e-commerce platforms, likely the best option. Localize accepts many common file types, including JSON, CSV and XML (Localize 2020a). Alternatively, Localize's content delivery system is an interesting feature that enables Localize to crawl the client website and extract all text content for translation. The text is automatically segmented based on HTML structure and imported into Localize. Once translated, Localize can also deliver the final content to the website and "listen" for changes, automatically loading new text into the application for translation (Localize 2020b). With this feature, a translation project is essentially managed end to end using Localize—though there exists a number of concerns with this approach.

Once imported, segments to be translated—termed "phrases" within the interface—are loaded into a "Pending" tab. From here, project managers can submit phrases for machine translation via Localize's integrations with Google Translate, Microsoft Translate and Amazon Translate. Alternatively, phrases can be assigned to human translators that have Localize accounts or exported for translation outside of the application entirely. Regardless of method, the segments move through a complete translation workflow, which includes in-progress, review and approval stages (see Figure 2).



FTRA-636: Localization Demo

Le Petit Prince

Lorsque j'avais six ans j'ai vu, une fois, une magnifique image, dans un livre sur la Forêt Vierge qui s'appelait"Histoires Vécues". Ça représentait un serpent boa qui avalait un fauve. Voilà la copie du dessin.

On disait dans le livre :"Les serpents boas avalent leur proie tout entière, sans la mâcher. Ensuite ils ne peuvent plus bouger et ils dorment pendant les six mois de leur digestion."

J'ai alors beaucoup réfléchi sur les aventures de la jungle et, à mon tour, j'ai réussi, avec un crayon de couleur, à tracer mon premier dessin. Mon dessin numéro i. Il était comme ça :

J'ai montré mon chef-d'oeuvre aux grandes personnes et je leur ai demandé si mon dessin leur faisait peur. Elles m'ont répondu :"Pourquoi un chapeau ferait-il peur ?"

Mon dessin ne représentait pas un chapeau. Il représentait un serpent boa qui digérait un éléphant. J'ai alors dessiné l'intérieur du serpent boa, afin que les grandes personnes puissent comprendre. Elles ont toujours besoin d'explications. Mon dessin numéro 2 était comme ça:

Les grandes personnes m'ont conseillé de laisser de côté les dessins de serpents boas ouverts ou fermés, et de m'intéresser plutôt à la géographie, à l'histoire, au calcul et à la grammaire. C'est ainsi que j'ai abandonné, à l'âge de six ans, une magnifique carrière de peintre. J'avais été découragé par l'insuccès de mon dessin numéro i et de mon dessin numéro 2. Les grandes personnes ne comprennent jamais rien toutes seules, et c'est fatigant, pour les enfants, de toujours et toujours leur donner des explications.

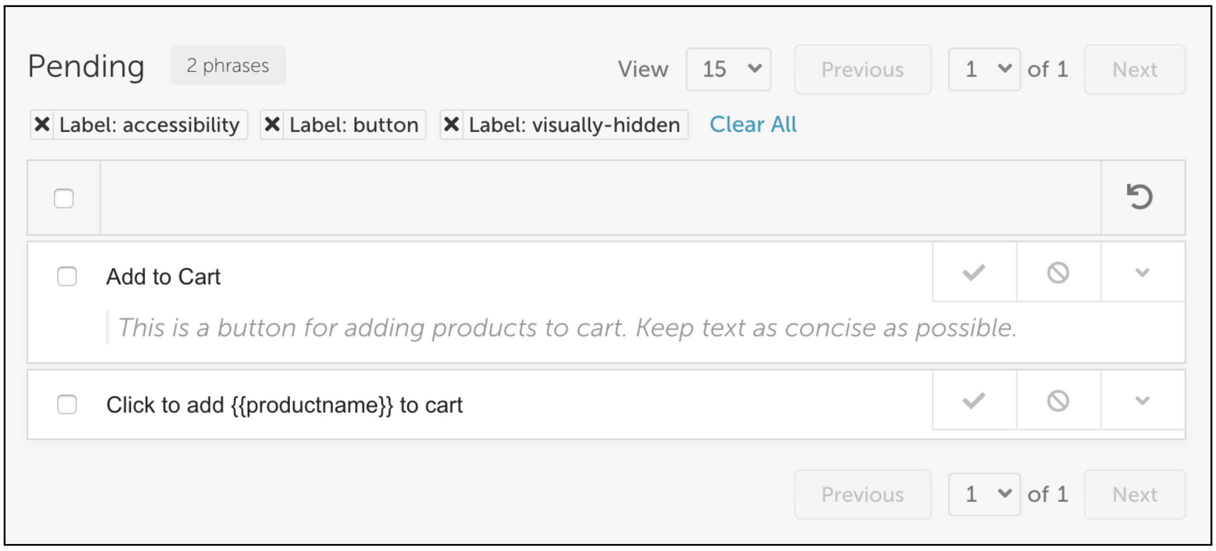
J'ai donc dû choisir un autre métier et j'ai appris à piloter des avions. J'ai volé un peu partout dans le monde. Et la géographie, c'est exact, m'a beaucoup servi. Je savais reconnaître, du premier coup d'oeil la Chine de l'Arizona. C'est très utile, si l'on est égaré pendant la nuit.

Figure 2: View of translation management workflow in Localize. Note that in this example, the imported French phrases that have yet to be translated and are available under the "Pending" tab.

Data tagging and contextualization

To optimize communication between disciplines, Localize allows developers to add context and additional information directly to text segments via a variety of means. First, labels can be applied to any text segment, which not only provide additional context but also enable Localize users to filter segments by label—a useful feature for projects with numerous text segments. Labels would also be helpful in flagging visually hidden ARIA content or providing translators with information about a text segment's functionality, such as whether the content appears in a button or is simply static. When labels aren't sufficient, programmers can provide more specific information to any segment via localization notes. These notes are easily added to any supported data file and will appear in tandem with the corresponding phrase in the Localize user interface for easy reference (see the first segment in Figure 3).

Perhaps most usefully, programmers can replace dynamic data with placeholder variable tags. Translators could be trained on the relatively simple syntax of variable tags, ensuring that the dynamic content within them doesn't get translated. This is especially important for projects leveraging Localize's content delivery system: if all dynamic content—such as usernames or product prices—is wrapped in variable tags, only static text will be available for translation. This could result in significant time savings by eliminating repetitive text strings in which only dynamic content differs (Localize 2020c) (see the second segment in Figure 3).



Pending 2 phrases View 15 Previous 1 of 1 Next

✗ Label: accessibility ✗ Label: button ✗ Label: visually-hidden Clear All

Add to Cart

This is a button for adding products to cart. Keep text as concise as possible.

Click to add {{productname}} to cart

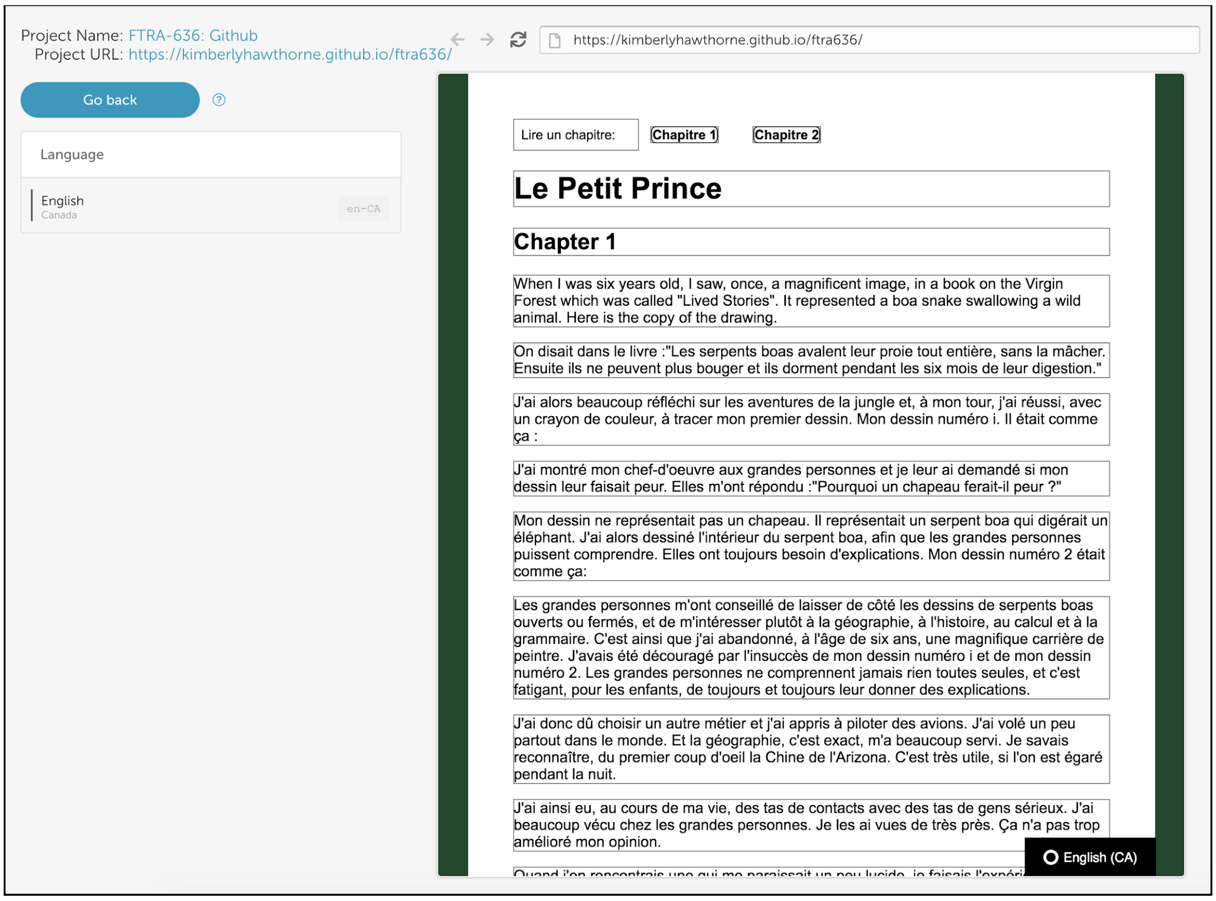
Previous 1 of 1 Next

Figure 3: Labels, notes and variables for dynamic data. In the first segment, a UX concern is noted. The second contains a variable for product name, which, without this variable tag, could be repeated for every product on the website, depending on import method.

```
{
   "labels": ["accessibility", "visually hidden"],
   "context": "Add localization notes here",
   "translation": {
      "Click to add <var productName>product</var> to cart": {}
   }
}
```

Figure 4: Example JSON format for a text segment with labels, notes and dynamic data.

Finally, clients using the content delivery feature have the benefit of previewing translations in the target website's user interface using Localize's in-site editor—essentially a what-you-see-is-what-you-get tool (see Figure 5). The in-site editor not only provides translators with real-world context for the segments they are translating, but also allows them to immediately detect any obvious UX or design issues that may result from translated output. Indeed, project managers could go so far as to add UX designers to Localize projects, enabling them to preview translated websites and approve them from a visual design perspective before they are published.



Project Name: FTRA-636: Github
Project URL: https://kimberlyhawthorne.github.io/ftra636/

https://kimberlyhawthorne.github.io/ftra636/

Go back

Language

English
Canada
en-CA

Lire un chapitre: Chapitre 1 Chapitre 2

Le Petit Prince

Chapter 1

When I was six years old, I saw, once, a magnificent image, in a book on the Virgin Forest which was called "Lived Stories". It represented a boa snake swallowing a wild animal. Here is the copy of the drawing.

On disait dans le livre :"Les serpents boas avalent leur proie tout entière, sans la mâcher. Ensuite ils ne peuvent plus bouger et ils dorment pendant les six mois de leur digestion."

J'ai alors beaucoup réfléchi sur les aventures de la jungle et, à mon tour, j'ai réussi, avec un crayon de couleur, à tracer mon premier dessin. Mon dessin numéro i. Il était comme ça :

J'ai montré mon chef-d'oeuvre aux grandes personnes et je leur ai demandé si mon dessin leur faisait peur. Elles m'ont répondu :"Pourquoi un chapeau ferait-il peur ?"

Mon dessin ne représentait pas un chapeau. Il représentait un serpent boa qui digérait un éléphant. J'ai alors dessiné l'intérieur du serpent boa, afin que les grandes personnes puissent comprendre. Elles ont toujours besoin d'explications. Mon dessin numéro 2 était comme ça:

Les grandes personnes m'ont conseillé de laisser de côté les dessins de serpents boas ouverts ou fermés, et de m'intéresser plutôt à la géographie, à l'histoire, au calcul et à la grammaire. C'est ainsi que j'ai abandonné, à l'âge de six ans, une magnifique carrière de peintre. J'avais été découragé par l'insuccès de mon dessin numéro i et de mon dessin numéro 2. Les grandes personnes ne comprennent jamais rien toutes seules, et c'est fatigant, pour les enfants, de toujours et toujours leur donner des explications.

J'ai donc dû choisir un autre métier et j'ai appris à piloter des avions. J'ai volé un peu partout dans le monde. Et la géographie, c'est exact, m'a beaucoup servi. Je savais reconnaître, du premier coup d'oeil la Chine de l'Arizona. C'est très utile, si l'on est égaré pendant la nuit.

J'ai ainsi eu, au cours de ma vie, des tas de contacts avec des tas de gens sérieux. J'ai beaucoup vécu chez les grandes personnes. Je les ai vues de très près. Ça n'a pas trop amélioré mon opinion.

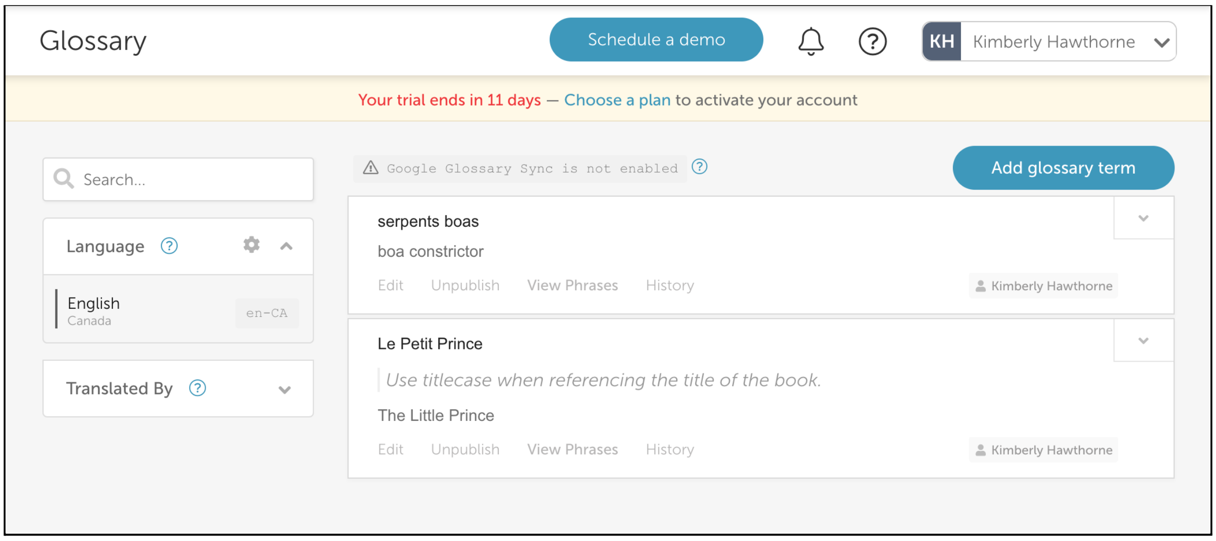
Quand j'en rencontrais une qui me paraissait un peu lucide, je faisais l'expéri

English (CA)

Figure 5: View of Localize's in-site editor

Enforcing consistency

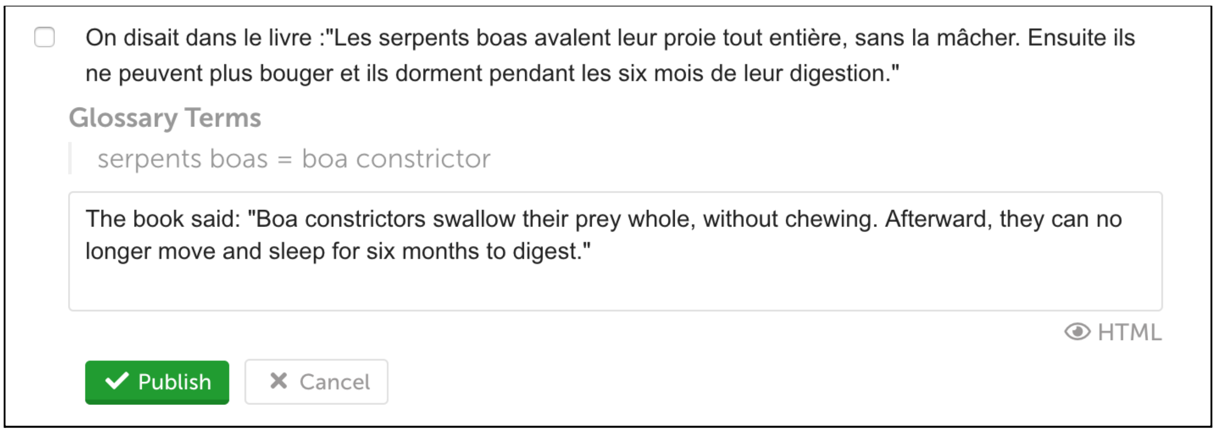
Localize has a number of tools available to project managers, translators and developers to ensure consistency in the localization process. When setting up a project, administrators can choose to add a style guide, which will be visible to all users of the application. Additionally, translators and/or terminology experts can create glossaries scoped to the project at hand. When translating segments that contain terms in the glossary, the entry appears in the translation user interface (see Figures 6 and 7).



serpents boas
boa constrictor

Le Petit Prince
Use titlecase when referencing the title of the book.
The Little Prince

Figure 6: Adding terms and notes to a glossary.



On disait dans le livre :"Les serpents boas avalent leur proie tout entière, sans la mâcher. Ensuite ils ne peuvent plus bouger et ils dorment pendant les six mois de leur digestion."

Glossary Terms

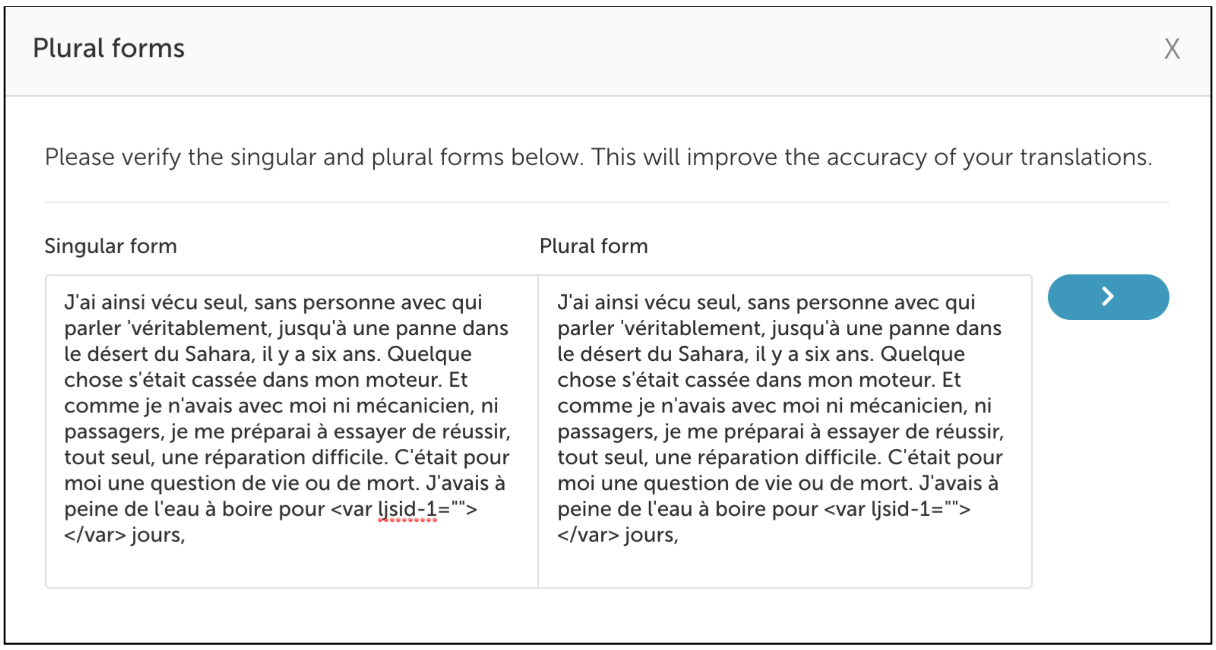
serpents boas = boa constrictor

The book said: "Boa constrictors swallow their prey whole, without chewing. Afterward, they can no longer move and sleep for six months to digest."

Figure 7: Localize interface for translating a text segment. Note that the glossary term for *serpent boas* is provided.

Lastly, Localize provides a unique solution via its content delivery system for handling singular and plural noun forms. Using a specific HTML tag, programmers can flag nouns as potentially needing to be pluralized. Once imported into Localize, translators are given the option to publish both singular and plural versions of the segment (see Figure 8). When the translations are finally served to the target website, the appropriate translated form—either singular or plural—is selected, depending on the integer defined in the HTML tag.

```
<p>You have <var pluralize="8">8</var> products in your cart</p>
```



Please verify the singular and plural forms below. This will improve the accuracy of your translations.

Singular form	Plural form
J'ai ainsi vécu seul, sans personne avec qui parler 'véritablement, jusqu'à une panne dans le désert du Sahara, il y a six ans. Quelque chose s'était cassée dans mon moteur. Et comme je n'avais avec moi ni mécanicien, ni passagers, je me préparai à essayer de réussir, tout seul, une réparation difficile. C'était pour moi une question de vie ou de mort. J'avais à peine de l'eau à boire pour <var ljsid-1=""></var> jours,	J'ai ainsi vécu seul, sans personne avec qui parler 'véritablement, jusqu'à une panne dans le désert du Sahara, il y a six ans. Quelque chose s'était cassée dans mon moteur. Et comme je n'avais avec moi ni mécanicien, ni passagers, je me préparai à essayer de réussir, tout seul, une réparation difficile. C'était pour moi une question de vie ou de mort. J'avais à peine de l'eau à boire pour <var ljsid-1=""></var> jours,

Figure 8: View of functionality to add both singular and plural forms of nouns. In this example, the final sentence contains a pluralize variable tag, and the word to be translated into singular and plural forms is *jour(s)*.

Evaluation of Localize's functionality

As detailed above, Localize offers project managers, translators and developers many features—some more useful than others—to streamline the process of localizing a website. This section will analyze the strengths and weaknesses of the tool.

Strengths

- Localize is generally easy to use, with an intuitive interface. Based on the author's experience, most project managers should be able to set up a project, add users with specific permissions and manage workflow without much training. When further clarification is needed, Localize provides thorough documentation.
- Importing and exporting data from Localize is simple, and the application supports many commonly used data file formats. The expected data structure for each file type is also clearly documented (Localize 2020a).

- The platform is not prescriptive from a workflow point of view, which means that teams can use it in whatever manner best fits their needs. Localize's content delivery feature manages all aspects of data import and delivery, which may be a good option for smaller businesses, bloggers or companies without access to Web developers.

  When using human translators, translation can be completed directly in the Localize application or the segments can be exported for external translation. In the latter scenario, Localize would simply act as a translation-tracking application, kept up to date by continuously importing both files that have yet to be translated and those that are already completed. Admittedly, this doesn't seem to be the best use of Localize; with its support for glossaries, dynamic placeholders and detailed revision history, Localize is best suited to managing translation workflows, with translations being completed directly in the application.
- Localize integrates with three major machine translation systems—Google, Microsoft and Amazon—to provide optional machine translations. Depending on a project's needs, these integrations could be leveraged to process raw translations, with human translators acting as post editors.
- The Localize platform offers many options for providing additional context to text segments, including labels, notes and style guides.
- Regardless of import method, Localize handles placeholders for dynamic content in an intuitive way. From a programmatic perspective, integrating the dynamic variable tags is simple and should have little to no impact on website performance.
- Localize easily handles projects with more than one target language.

Weaknesses

- Localize's content delivery system is the source of all major concerns, though it's quite heavily promoted by the company. While the system's ability to fetch data and segment it based on HTML structure is indeed quite useful, there exist two major issues with this approach from a programmatic perspective.

  First, in order to register content with the delivery system, a script has to be added to the website, after which each page of the website must be loaded. This is problematic for single-page applications (SPA), which are becoming increasingly popular. A SPA is loaded only once; all other interactions and navigation are handled on the front end using JavaScript (Microsoft Docs 2019). Additionally, many SPA websites are heavily dynamic and tailored to users. They rely on data coming from a server via an API, which itself might contain textual content that requires localization. Without the ability to explicitly load all of these experiences, it's possible that Localize would never register them and, in turn, never translate them. Conversely, Localize does offer an API integration; however, such a solution would likely make no difference for the content delivery system and was outside the scope of this research.

  However, let's assume that Localize *is* able to fetch all of a website's textual content for translation. In this scenario, the translated segments would be delivered back to the target website on the client side—that is, the website will first load, then request the translated content from Localize and finally display it. This can have serious implications for search engine optimization (SEO), which is "a methodology of strategies, techniques and tactics used to increase the amount of visitors (traffic) to a website by obtaining a high-ranking placement in the search results page of a search engine" (Beal n.d.). Generally, website content is crawled and indexed by search engines on the server; thus, with translations being rendered only on the client, SEO may be heavily impacted for a website that uses Localize's content delivery system. This alone could be a dealbreaker for companies that rely heavily on search engine results for traffic.

- Localize’s website mentions translation memories as a feature, but neither the user interface nor the documentation indicate such support.
- Finally, Localize doesn’t support pluralization in imported text segments. Developers can still wrap nouns to be pluralized in a variable—essentially marking them as dynamic content—but Localize’s singular/plural functionality will not be available.

Conclusion

Web localization is, by nature, a complex process. It entails much more than language transfer, as programmatic, design and accessibility concerns must also be considered. In general, Localize is a solid localization management offering. Its principle weakness concerns the client-side content delivery system and the programmatic implications of using it. However, Localize’s core functionality—managing translations and localization workflow—would certainly enhance communication between professionals in different disciplines and ensure better consistency in a large-scale localization project. Indeed, as the Web continues to grow, so will the necessity for solutions, like Localize, that streamline this process.

Bibliography

Beal, Vangie. "Search engine optimization." Webopedia. Accessed April 13, 2020. https://www.webopedia.com/TERM/S/SEO.html

De Sainte-Exupéry, Antoine. "Le Petit Prince." 1945. Accessed on April 13, 2020. http://lepetitprinceexupery.free.fr (Note that this data was used to publish a website to test Localize)

Globalization and Localization Association. n.d. "What is Localization?" Accessed April 13, 2020. https://www.gala-global.org/industry/intro-language-industry/what-localization

Hawthorne, Kimberly. "The Intersection of Localization and Web Accessibility for Visually Impaired Users." Paper submitted for FTRA-600 at Concordia University. December 3, 2019. Available on demand.

Internet World Stats. Last modified March 26, 2020. "Internet World Users By Language." Accessed April 12, 2020. https://www.internetworldstats.com/stats7.htm

Jiménez-Crespo, Miguel A. 2013. Translation and Web Localization. Abingdon, Oxon: Routledge.

Localize. 2020a. "Getting Content Into Localize." Last modified January 2020. https://help.localizejs.com/docs/installation

Localize. 2020b. "How Translations Get to Your Website." Last modified January 2020. https://help.localizejs.com/docs/how-translations-get-to-your-website

Localize. 2020c. "Using HTML Syntax to Prepare Phrases." Last modified January 2020. https://help.localizejs.com/docs/setup-guide

MDN Web Docs. "An overview of accessible web applications and widgets." Last modified June 19, 2019. https://developer.mozilla.org/en-US/docs/Web/Accessibility/An_overview_of_accessible_web_applications_and_widgets

MDN Web Docs. "Localization content best practices." Last modified February 18, 2020. https://developer.mozilla.org/en-US/docs/Mozilla/Localization/Localization_content_best_practices

Microsoft Docs. "Choose Between Traditional Web Apps and Single Page Apps (SPAs)." December 4, 2019.

https://docs.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/choose-between-traditional-web-and-single-page-apps

W3C Internationalization. 2016a. “Text size in translation.” Last modified February 1, 2016. https://www.w3.org/International/articles/article-text-size

W3C Internationalization. 2016b. “Internationalization techniques: Authoring HTML & CSS.” Last modified January 28, 2016. https://www.w3.org/International/techniques/authoring-html

W3C Web Accessibility Initiative. “Tools and Techniques.” Last modified May 15, 2017. https://www.w3.org/WAI/people-use-web/tools-techniques/